StorageTek®

PART NUMBER
312603501

VERSION NUMBER
4.2

EDITION NUMBER
1

ASM

APPLICATION STORAGE MANAGER® (ASM) SOFTWARE

ASM AND QFS FILE SYSTEM ADMINISTRATION GUIDE

For Solaris

PRODUCT TYPE
SOFTWARE

StorageTek® QFS and ASM™

File System Administration Guide

Information contained in this publication is subject to change without notice.

We welcome your feedback. Please contact the Global Learning Solutions Feedback System at:

GLSFS@Stortek.com

or

Global Learning Solutions
Storage Technology Corporation
One StorageTek Drive
Louisville, CO 80028-3256
USA

Please include the publication name, part number, and edition number in your correspondence if they are available.

Export Destination Control Statement

These commodities, technology or software were exported from the United States in accordance with the Export Administration Regulations. Diversion contrary to U.S. law is prohibited.

Restricted Rights

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) and (2) of the Commercial Computer Software - Restricted Rights at FAR 52.227-19 (June 1987), as applicable.

Limitations on Warranties and Liability

Storage Technology Corporation cannot accept any responsibility for your use of the information in this document or for your use in any associated software program. You are responsible for backing up your data. You should be careful to ensure that your use of the information complies with all applicable laws, rules, and regulations of the jurisdictions in which it is used.

Proprietary Information Statement

The information in this document, including any associated software program, may not be reproduced, disclosed or distributed in any manner without the written consent of Storage Technology Corporation.

Should this publication be found, please return it to StorageTek, One StorageTek Drive, Louisville, CO 80028-5214, USA. Postage is guaranteed.

First Edition (November 2004)

Contents

Preface

This manual, the *StorageTek QFS and ASM File System Administration Guide*, describes the file system software included in the StorageTek QFS and StorageTek ASM 4.2 releases. The software products and the file systems they include are as follows:

- StorageTek ASM file system. The StorageTek ASM environment includes a general-purpose file system along with the Application Storage Manager, ASM. The StorageTek ASM environment's file system enables data to be archived to automated libraries at device-rated speeds. Data can also be archived to files in another file system through a process known as *disk archiving*. The file system in the StorageTek ASM environment is a complete file system. The user is presented with a standard file system interface and can read and write files as though they were all on primary disk storage.
- StorageTek QFS file system. The StorageTek QFS file system can be used as a standalone file system, or it can be used in conjunction with the Application Storage Manager, ASM. When used in conjunction with ASM, it is known as *StorageTek ASM-QFS*. StorageTek QFS shares most of the StorageTek ASM file system's features. The StorageTek QFS file system, however, is designed for high performance and contains more features than are supported within the StorageTek ASM environment.

Note: You can purchase licenses for both StorageTek QFS and StorageTek ASM software with the intent to run the StorageTek QFS file system with the storage and archive manager found in the StorageTek ASM software. Such a system is referred to as *StorageTek ASM-QFS*.

This manual does not call out the StorageTek ASM-QFS configuration unless it is necessary for clarity. In this manual, you can assume that references to StorageTek ASM also apply to StorageTek ASM-QFS configurations when talking about storage and archive management. Likewise, you can assume that references to StorageTek QFS also apply to StorageTek ASM-QFS configurations when talking about file system design and capabilities.

The StorageTek QFS and StorageTek ASM file systems are technologically similar, but within this manual, differences are noted when necessary.

The StorageTek QFS and StorageTek ASM 4.2 releases are supported on the following minimum Sun Solaris™ Operating System (OS) platform levels.

Table i. Minimum Sun Solaris OS Platform Levels

Product	Minimum Platform Levels
StorageTek QFS	Solaris 8 07/01
	Solaris 9 04/03
StorageTek ASM	Solaris 8 07/01
	Solaris 9 04/03
StorageTek QFS in a Sun Cluster environment	Solaris 8 02/02
	Solaris 9 04/03

This manual is written for system administrators responsible for installing, configuring, and maintaining StorageTek QFS and StorageTek ASM file systems. You, the system administrator, are assumed to be knowledgeable about Solaris OS procedures, including installation, configuration, creating accounts, performing system backups, and other basic Solaris OS system administration tasks.

How This Book Is Organized

This manual contains the following chapters:

- Chapter 1 provides overview information.
- Chapter 2 provides file system design information.
- Chapter 3 provides volume management information.
- Chapter 4 explains how to perform various tasks for the StorageTek QFS and StorageTek ASM file systems. Tasks covered include initializing a file system, adding a server, adding disk cache, and other system administration activities.
- Chapter 5 explains how to use the shared StorageTek QFS file system.
- Chapter 6 explains how to use the `samu`(1M) operator utility.
- Chapter 7 explains how to use file system quotas.
- Chapter 8 describes how StorageTek QFS software works in a Sun Cluster environment.
- Chapter 9 explains miscellaneous advanced topics such using multireader file system and performance features.

The glossary defines terms used in this and other StorageTek QFS and StorageTek ASM documentation.

Using UNIX Commands

This document does not contain information on basic UNIX® commands and procedures such as shutting down the system, booting the system, and configuring devices. Refer to one or more of the following for this information:

- Software documentation that you received with your system
- Solaris OS documentation, which is at the following URL:

  http://docs.sun.com

Shell Prompts

Table ii. shows the shell prompts that this manual uses.

Table ii. Shell Prompts

Shell	Prompt
C shell	*machine-name*%
C shell superuser	*machine-name*#
Bourne shell and Korn shell	$
Bourne shell and Korn shell superuser	#

Typographic Conventions

Table iii. lists the typographic conventions used in this manual.

Table iii. Typographic Conventions

Typeface or Symbol	Meaning	Examples
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output.	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`
`AaBbCc123`	What you type, when contrasted with on-screen computer output.	`% su` `Password:`
AaBbCc123	Book titles; new words or terms; words to be emphasized; and command line variables to be replaced with a real name or value.	Read Chapter 6 in the *User's Guide*. These are called *class* options. You *must* be `root` to do this. To delete a file, type `rm` *filename*.
[ ]	In syntax, brackets indicate that an argument is optional.	`scmadm` [`-d` *sec*] [`-r` *n*[:*n*][,*n*]...] [`-z`]
{ *arg* | *arg* }	In syntax, braces and pipes indicate that one of the arguments must be specified.	`sndradm -b` { *phost* | *shost* }
\	At the end of a command line, the backslash (\) indicates that the command continues on the next line.	`atm90 /dev/md/rdsk/d5 \` `/dev/md/rdsk/d1 atm89`

■ ASM for Solaris Documentation

This section lists software and hardware documentation for the ASM for Solaris products.

How to Obtain ASM for Solaris Documentation

All ASM for Solaris publications are available from the following sources:

- On the ASM for Solaris CD-ROM. To order a copy, contact StorageTek Publication Sales and Service at 800.436.5554 or send a fax to 303.661.7367.
- Online (for viewing and printing), at the StorageTek Customer Resource Center (CRC) website at: `www.support.storagetek.com`. To access the ASM for Solaris publications, use these steps:

    a. Log in.

    Note: Logging in requires a customer login ID and password which can be obtained by calling StorageTek Customer Support at 800-678-4430.

    b. Click **Product Information**.

    c. Click **Current Products**.

    d. Click **Software** from the side menu.

    e. Scroll to **Storage Management Software**.

    f. Click **ASM for Solaris**.

    The ASM for Solaris publications are available under the “Manuals and Guides” heading. Click **View More** to see the complete list.

ASM for Solaris Software Publications

The ASM for Solaris software library consists of:

- *ASM Storage and Archive Manager Guide*, part number 312603201
- *ASM-Remote Administrator’s Guide*, part number 312603301
- *ASM Installation and Configuration Guide*, part number 312603401
- *QFS and ASM File System Administration Guide*, part number 312603501
- *ASM/QFS Disaster Recovery Guide*, part number 312603601
- *QFS and ASM Release Notes*, part number 312603801

Related Publications

For any StorageTek software:

- *Requesting Help from Software Support*, part number 11212400*n*

Licensing

For information on obtaining licenses for StorageTek QFS and StorageTek ASM software, contact your StorageTek sales representative or your authorized service provider.

Diagnostics

The StorageTek QFS and StorageTek ASM software includes the `samexplorer`(1M) script. This diagnostic script can be very useful to you and to the StorageTek customer support staff. This script produces a diagnostic report of the server configuration and collects log information. After the software is installed, you can access the `samexplorer`(1M) man page for more information about this script.

Overview

1

The StorageTek QFS and StorageTek ASM file systems are configurable file systems that present a standard UNIX file system interface to users. Table 1. shows how these file systems can be used or combined with the Application Storage Manager (ASM).

Table 1. Product Overview

Product	Components
StorageTek QFS file system	StorageTek QFS standalone file system
StorageTek ASM file system	Standard file system plus the Application Storage Manager, ASM
StorageTek ASM-QFS file system	The StorageTek QFS file system combined with the storage and archive management utilities found in the StorageTek ASM software.

Technologically, the two file systems are similar, but there are also differences between them. This chapter presents an overview of the features common to these file systems, highlights the features that differentiate the file systems, and explains the commands available with each file system. Specifically, this chapter is divided into the following sections:

- "Common Features" on page 1
- "File System Differences" on page 4

Common Features

The StorageTek QFS and StorageTek ASM file systems do not require changes to user programs, nor are changes required to the UNIX kernel. These file systems share the features described in the following sections.

vnode Interface

The StorageTek QFS and StorageTek ASM file systems are implemented using the standard Solaris operating system (OS) virtual file system (`vfs/vnode`) interface.

By using the `vfs/vnode` interface, these file systems work with the standard Solaris OS kernel and require no modifications to the kernel for file management support. Thus, the file system is protected from operating system changes and typically does not require extensive regression testing when the operating system is updated.

The kernel intercepts all requests for files, including those that reside in StorageTek QFS and StorageTek ASM file systems. If the file is identified as a StorageTek QFS or StorageTek ASM file, the kernel passes the request to the appropriate file system for handling. StorageTek QFS and StorageTek ASM file systems are identified as type `samfs` in the `/etc/vfstab` file and on the `mount`(1M) command.

Enhanced Volume Management

StorageTek QFS and StorageTek ASM file systems support both striped and round-robin disk access. The master configuration file (`mcf`) and the mount parameters specify the volume management features and let the file system know the relationships between the devices it controls. This is in contrast to most UNIX file systems that can address only one device or one portion of a device. StorageTek QFS and StorageTek ASM file systems do not require any additional volume management applications. If you want to use mirroring for any devices in a StorageTek QFS or StorageTek ASM environment, obtain an additional package, such as a logical volume manager.

The StorageTek QFS and StorageTek ASM integrated volume management features use the standard Solaris OS device driver interface to pass I/O requests to and from the underlying devices. The StorageTek QFS and StorageTek ASM software groups storage devices into family sets upon which each file system resides.

Support for Paged and Direct I/O

StorageTek QFS and StorageTek ASM file systems support two different types of I/O: paged (also called *cached* or *buffered* I/O) and direct. These I/O types are as follows:

- When paged I/O is used, user data is cached in virtual memory pages and the kernel writes the data to disk. The standard Solaris OS interfaces manage paged I/O. This is the default type of I/O.
- When direct I/O is used, user data is written directly from user memory to disk. You can specify direct I/O by using the Solaris OS `directio`(3C) function call or the `setfa`(1) command with its `-D` option. Large block, sequential, aligned I/O can realize substantial performance improvements by using direct I/O.

Preallocation of File Space

You can use the `setfa`(1) command to preallocate contiguous disk space for fast sequential reads and writes.

Application Programming Interface Routines

The application programming interface (API) routines enable a program to perform various specialized functions, such preallocating contiguous disk space or accessing a specific striped group. For more information about these routines, see the `intro_libsam`(3) man page.

High Capacity

The StorageTek QFS and StorageTek ASM file systems support files of up to 2^{63} bytes in length. Such very large files can be striped across many disks or RAID devices, even within a single file system. This is true because StorageTek QFS and StorageTek ASM file systems use true 64-bit addressing. This is in contrast to a standard UNIX file system (UFS), which is not a true 64-bit file system.

The number of file systems you can configure is virtually unlimited. The volume manager enables each file system to include up to 252 device partitions (typically disk). Each partition can include up to 4 terabytes of data. This configuration offers virtually unlimited storage capacity.

There is no predefined limit on the number of files on a StorageTek ASM file system. Because the inode space (which holds information about the files) is dynamically allocated, the maximum number of files is limited only by the amount of disk storage available. The inodes are cataloged in the `.inodes` file under the mount point. The `.inodes` file requires 512 bytes of storage per file.

For a StorageTek QFS file system, the inodes are located on the metadata device(s) and are separated from the file data devices. In practice, the size of your metadata (`mm`) devices sets the limit on the number of files in a StorageTek QFS file system. You can increase maximum the number of files by adding more metadata devices. The hard limit on the number of files is 2^{32}-1 files, and the recommended limit is 10^7 files.

Fast File System Recovery

A key function of a file system is the ability to recover quickly after an unscheduled outage. Standard UNIX file systems require a lengthy file system check (`fsck`(1M)) to repair inconsistencies after a system failure.

StorageTek QFS and StorageTek ASM file systems often do not require file system checks after a disruption that prevents the file system from being written to to disk (using `sync`(1M)). In addition, they recover from system failures without using journaling. They accomplish this dynamically by using identification records, serial writes, and error checking for all critical I/O operations. After a system failure, StorageTek QFS and StorageTek ASM file systems can be remounted immediately, even for multiterabyte-sized file systems.

Adjustable Disk Allocation Unit

The disk allocation unit (DAU) is the basic unit of online storage. The StorageTek QFS file systems include an adjustable DAU, which is useful for tuning the file system with the physical disk storage device and for eliminating the system overhead caused by read-modify-write operations. You can adjust the DAU size in multiples of 4 kilobytes.

■ File System Differences

The StorageTek QFS and StorageTek ASM file systems share many features, and these are described in “Common Features” on page 1. This section, however, describes the areas in which they differ. One area of difference is performance. The StorageTek QFS file system provides the ability to attain raw, device-rated disk speeds with the administrative convenience of a file system. The following sections note other ways in which the file systems differ.

Metadata Storage

File systems use metadata to reference file and directory information. Typically, metadata resides on the same device as the file data. This is true for the StorageTek ASM file system.

The StorageTek QFS file system separates the file system metadata from the file data by storing them on separate devices. The StorageTek QFS file system enables you to define one or more separate metadata devices in order to reduce device head movement and rotational latency, improve RAID cache utilization, or mirror metadata without mirroring file data.

Both the StorageTek QFS and StorageTek ASM file systems store inode metadata information in a separate file. This enables the number of files, and the file system as a whole, to be enlarged dynamically.

Support for Multiple Striped Groups

To support multiple RAID devices in a single file system, striped groups can be defined in StorageTek QFS file systems. You can optimize disk block allocation for a striped group. This reduces the overhead for updating the on-disk allocation map. Users can assign a file to a striped group either through an API routine or by using the `setfa`(1) command.

ASM Interoperability

The StorageTek ASM file system combines file system features with the Application Storage Manager, ASM. Users can read and write files directly from magnetic disk, or they can access archive copies of files as though they were all on primary disk storage.

The StorageTek QFS file system can be used as a standalone file system, or it can be used in conjunction with the Application Storage Manager, ASM. If you are licensed for both StorageTek QFS and StorageTek ASM, it is called *StorageTek ASM-QFS*.

When possible, StorageTek ASM sofware uses the standard Solaris OS disk and tape device drivers. For devices not directly supported under the Solaris OS, such as certain automated library and optical disk devices, StorageTek provides special device drivers in the StorageTek ASM software package.

Sun Cluster Interoperability

The StorageTek QFS file system is supported as a local file system and as a highly available file system in a Sun Cluster environment. The StorageTek ASM file system is not supported in a Sun Cluster environment.

StorageTek QFS Shared File System Support

The shared file system can be implemented as either a StorageTek QFS shared file system or as a StorageTek ASM-QFS shared file system. A shared file system enables you to implement a distributed file system that can be mounted on multiple Sun Solaris host systems.

StorageTek QFS shared file systems do not support the following file types:

- `b` — block special files
- `c` — character special files
- `p` — FIFO (named pipe) special files

The shared file system does not support segmented files. You cannot implement a StorageTek ASM-QFS shared file system in a Sun Cluster environment.

For more information about this file system, see the “StorageTek QFS Shared File System” on page 85.

File System Design

2

Well-designed file systems are critical to ensuring quick and uninterrupted access to information. Good design is also essential to file system recovery. This chapter contains the following sections:

Design Basics

StorageTek QFS and StorageTek ASM file systems are multithreaded, advanced storage management systems. To take maximum advantage of these capabilities, create multiple file systems whenever possible.

The StorageTek QFS and StorageTek ASM file systems use a linear search method when performing directory lookups. They search from the beginning of the directory to the end. As the number of files in a directory increases, the search time through the directory also increases. Users who have directories with thousands of files can experience excessive search times. These long search times are also evident when you restore a file system. To increase performance and speed up file system dumps and restores, you should keep the number of files in a directory under 10,000.

Both the directory name lookup cache (DNLC) feature and the directory DNLC feature improve file system performance. Directory DNLC is available in all Solaris operating system (OS) 9 releases and in the later Solaris OS release 8 updates.

Inode Files and File Characteristics

The types of files to be stored in a file system affect file system design. An *inode* is a 512-byte block of information that describes the characteristics of a file or directory. This information is allocated dynamically within the file system.

The inodes are stored in the `.inodes` file located under the file system mount point. A StorageTek ASM `.inodes` file resides on the same physical device as the file data and is interleaved with the file data. In contrast, a

StorageTek QFS `.inodes` file resides on a metadata device that is separate from the file data device.

Like a standard Solaris operating system (OS) inode, a StorageTek QFS or StorageTek ASM file system inode contains the file's POSIX standard inode times: file access, file modification, and inode changed times. The StorageTek QFS and StorageTek ASM file systems add a creation time, an attribute change time, and a residence time. Table 2. summarizes the times that are recorded in the inode.

Table 2. Content of `.inode` Files

Time	Incident
`access`	Time the file was last accessed. POSIX standard.
`modification`	Time the file was last modified. POSIX standard.
`changed`	Time the inode information was last changed. POSIX standard.
`attributes`	Time the attributes specific to the StorageTek QFS or StorageTek ASM files systems were last changed. StorageTek extension.
`creation`	Time the file was created. StorageTek extension.
`residence`	Time the file changed from offline to online or vice versa. StorageTek extension.

The attributes specific to the StorageTek QFS and StorageTek ASM file systems include both user settings and general file states. The following two sections describe these characteristics.

File Attributes and File States

A file's user-specified attributes and its system-specified states are stored in the file's inode. You can use the `sls`(1) `-D` command to display these inode attributes. For more information about `sls`(1) options, see the `sls`(1) man page.

A user can specify the following commands to set attributes:

- `archive`(1)
- `ssum`(1)
- `release`(1)
- `segment`(1)
- `setfa`(1)
- `stage`(1)

Users can set attributes from within applications by specifying the following application programming interface (API) routines:

- `sam_archive`(3)
- `sam_release`(3)
- `sam_segment`(3)
- `sam_setfa`(3)
- `sam_ssum`(3)
- `sam_stage`(3)

User-Specified File Attributes

Table 3. shows the user-specified attributes that are listed in the inode.

Table 3. User-Specified File Attributes

Command	Definition	Used By
`archive -C`	The file is marked for concurrent archiving. This means that the file can be archived even if it is open for a write operation. You can use the `archive`(1) command to set this attribute.	StorageTek ASM
`archive -n`	The file is marked to never be archived. The superuser can use the `archive`(1) command to set this attribute.	StorageTek ASM
`release -a`	This file is marked to be released as soon as one archive copy is made. You can set this attribute from within the `archiver.cmd` file or by using the `release`(1) command.	StorageTek ASM
`release -n`	This file is marked to never be released. You can set this attribute from within the `archiver.cmd` file, or the superuser can use the `release`(1) command to set it.	StorageTek ASM
`release -p`	The file is marked for partial release. You can set this attribute from within the `archiver.cmd` file or by using the `release`(1) command.	StorageTek ASM

Table 3. User-Specified File Attributes (Continued)

Command	Definition	Used By
`stage -a`	The file is marked for associative staging. You can set this attribute from within the `archiver.cmd` file or by using the `stage`(1) command.	StorageTek ASM
`stage -n`	The file is marked to never be staged. This signifies direct access to removable media cartridges. You can set this attribute from within the `archiver.cmd` file, or the superuser can use the `stage`(1) command to set it. Not supported on StorageTek QFS shared file system clients.	StorageTek ASM
`setfa -D`	The file is marked for direct I/O.	StorageTek QFS StorageTek ASM
`setfa -g`*n*	The file is marked for allocation on striped group *n*.	StorageTek QFS
`setfa -s`*m*	The file is marked for allocation with a stripe width of *m*.	StorageTek QFS StorageTek ASM
`segment` *n*`m` `stage_ahead` *x*	The file is marked for segmentation. The *n*`m` notation indicates that the segment is *n* megabytes in size. The `stage_ahead` *x* attribute indicates the number of segments (*x*) to be staged ahead. You can use the `segment`(1) command to set this attribute.	StorageTek ASM

You can set the attributes shown in Table 3. on both files and directories. After directory attributes are set, files that are created in the directory inherit all the directory attributes at the time of creation. Files created before an attribute is applied to the parent directory do not inherit directory attributes.

Users can gather information about file attributes by using the `sls`(1) command, which is described in “Displaying File Information” on page 11.

System-Specified File States

Table 4. shows the various states that the file systems set for a file. These states are stored in the inode.

Table 4. System-Specified File States

Attribute	Definition	Used By
`archdone`	Indicates that the file's archive requirements have been met. There is no more work the archiver must do on the file. The archiver sets this attribute. It cannot be set by a user. Note that `archdone` does not necessarily indicate that the file has been archived.	StorageTek ASM
`damaged`	The file is damaged. The stager or the `samfsrestore`(1M) command sets this attribute. You can use the `undamage`(1M) command to reset this attribute to undamaged. If this attribute has been set by the `samfsrestore`(1M) utility, it means that no archive copies existed for the file at the time a `samfsdump`(1M) was taken. You can reset this attribute to undamaged, but the file might still be unrecoverable.	StorageTek ASM
`offline`	The file data has been released. The releaser sets this attribute. You can also set this attribute by using the `release`(1) command.	StorageTek ASM

Users can gather information about file states by using the `sls`(1) command, which is described in "Displaying File Information" on page 11.

Displaying File Information

The StorageTek QFS and StorageTek ASM `sls`(1) command extends the standard UNIX `ls`(1) command and provides more information about a file.

Figure 1. shows detailed `sls`(1) command output that displays the inode information for file `hgc2`.

Figure 1. `sls`**(1) Output in a StorageTek ASM Environment**

```
# sls -D hgc2
hgc2:
  mode: -rw-r--r--  links:   1  owner: root     group: other
  length:     14971  admin id:      0  inode:       30.5
  archdone;
  segments 3, offline 0, archdone 3, damaged 0;
  copy 1: ---- Jun 13 17:14      2239a.48   lt MFJ192
  copy 2: ---- Jun 13 17:15       9e37.48   lt AA0006
  access:      Jun 13 17:08  modification: Jun 13 17:08
  changed:     Jun 13 17:08  attributes:   Jun 13 17:10
  creation:    Jun 13 17:08  residence:    Jun 13 17:08
```

Table 5. describes the meaning of each row of `sls`(1) output shown in Figure 1. In Table 5., note that lines that pertain to archiving do not appear in `sls`(1) output in a StorageTek QFS environment.

Table 5. `sls`**(1) Output Explanation**

Line Number	First Few Characters	Content
1	`mode:`	The file's mode and permissions, the number of hard links to the file, the owner of the file, and the group to which the owner belongs.
2	`length:`	The file's length in bytes, the file's admin ID number, and the file's inode number. By default, the admin ID number is 0. If this number is greater than 0, it indicates the file's accounting category for counting files and blocks. You can set this number to a value greater than 0 even when file system quotas are not enabled on this file system. For information about file system quotas, see "File System Quotas" on page 199. The inode number is a two-part number that contains the inode number itself, followed by a period (.), followed by the inode generation number.
3	`archdone;`	The file attributes specific to the file. For more information about this line, see the `sls`(1) man page.

Table 5. `sls`**(1) Output Explanation (Continued)**

Line Number	First Few Characters	Content
4	`segments`	The segment index information. This line does not appear unless the file is a segment index. The general format for this line is as follows: `segments` *n*`, offline` *o*`, archdone` *a*`, damaged` *d*`;` `segments` *n* shows the total number of data segments for this file. In this example, there are 3. `offline` *o* shows the number of data segments offline. In this example, there are no offline segments. `archdone` *a* shows the number of segments for which the archiving requirements have been met. In this example, there are 3. `damaged` *d* shows the number of damaged segments. In this example, there are no damaged segments.
5	`copy 1:`	The first archive copy line. The `sls`(1) command displays one archive copy line for each active or expired archive copy. For more information, see "Archive Copy Line Explanation" on page 13.
6	`copy 2:`	The second archive copy line. For more information, see "Archive Copy Line Explanation" on page 13.
7	`access:`	The time the file was last accessed and modified.
8	`changed:`	The time the file content was last changed and since the file's attributes were last changed.
9	`creation:`	The time the file was created and became resident in the file system.

Archive Copy Line Explanation

The fields in the archive copy lines are as follows:

- The first field indicates the archive copy number.

- The second field contains four indicators, each of which is either a dash (-) or a letter. Reading them from left to right, Table 6. shows the information that the indicators convey.

Table 6. Archive Copy Line Indicators

Position	Meaning
1	Indicates either an expired or active entry. An `S` indicates that the archive copy is expired. That is, the file was modified and this archive copy is a previous version of the file. A `U` indicates that the copy has been unarchived. *Unarchiving* is the process by which archive entries for files or directories are deleted. A dash (-) indicates that the archive copy is active and valid.
2	Indicates whether the archive copy is to be rearchived. An `r` indicates that the archive copy is scheduled to be rearchived by the archiver. A dash (`-`) indicates that the archive copy is not to be rearchived by the archiver.
3	Unused.
4	Indicates whether the copy is damaged or undamaged. A `D` indicates that the archive copy is damaged. A damaged archive copy is not a candidate for staging. A dash (-) indicates that the archive copy is not damaged. It is a candidate for staging.

- The third field shows the date and time the archive copy was written to the archive media.
- The fourth field contains two hexadecimal numbers separated by a decimal point (.). The first hexadecimal number (`2239a`) indicates the position of the beginning of the archive file on the cartridge. The second hexadecimal number (`48`) is the file byte offset (divided by 512) of this copy in the archive file.
- The fifth and sixth fields in the archive copy line indicate the media type and the Volume Serial Name (VSN) where the archive copy resides.

Checksum Line Explanation

If a file has checksum-related attributes, the `sls`(1) command returns a `checksum` line. You can use the `ssum`(1) command to set these attributes (`generate`, `use`, or `valid`). This line appears in `sls`(1) output in StorageTek ASM environments. The format of the checksum line is as follows:

```
checksum: gen  use  val  algo:  1
```

The system displays the preceding line if checksum attributes are set for a file. You can interpret this line as follows:

- If the `generate` attribute is not set, `no_gen` appears in place of `gen`.
- If the `use` attribute is not set, `no_use` appears.
- If the file has been archived and a checksum has been computed, `val` appears.
- If the file has not been archived or if no checksum has been computed, `not_val` appears.
- The keyword `algo` precedes the numeric algorithm indicator that specifies the algorithm that is used to generate the checksum value.

Specifying Disk Allocation Units and Stripe Widths

Disk space is allocated in blocks. These are also called *disk allocation units* (DAUs), which are the basic units of online disk storage. While sectors, tracks, and cylinders describe the physical disk geometry, the DAU describes the file system geometry. Choosing the appropriate DAU size and stripe size can improve performance and optimize magnetic disk usage. The DAU setting is the minimum amount of contiguous space that is used when a file is allocated.

Example: Assume that you have a StorageTek ASM file system. Your DAU is set to 16 kilobytes and you have disabled striping by setting `stripe=0`. You are using round-robin allocation (because of the `stripe=0` setting), and you have two files, as follows:

- The first file is a 15-kilobyte file. It occupies one DAU. The file data occupies 15 kilobytes of the DAU, and the other 1 kilobyte is not used.
- The second file is a 20-kilobyte file. It occupies two DAUs. The file data occupies all 16 kilobytes of the first DAU, and 4 kilobytes of the second DAU. The second DAU contains 12 kilobytes that are not used.

The `-a` *allocation_unit* option on the `sammkfs`(1M) command specifies the DAU setting.

If striped allocation is used, the stripe width mount option determines the maximum number of DAUs written in one I/O event. This setting is specified by the `-o stripe=`*n* option on the `mount`(1M) command. You must run the `sammkfs`(1M) command before you run the `mount`(1M) command.

The following sections describe how to configure DAU settings and stripe widths.

Note: Unless otherwise noted, StorageTek QFS *file system* information throughout this manual applies to StorageTek ASM-QFS configurations as well.

DAU Settings and File System Geometry

The StorageTek QFS and StorageTek ASM file systems use an adjustable DAU. You can use this configurable DAU to tune the file system to the physical disk storage device. This minimizes the system overhead caused by read-modify-write operations. Applications that manipulate very large files can benefit substantially from this feature. For information about how to control the read-modify-write operation, see "Increasing Large File Transfer Performance" on page 299.

Each file system can have its own unique DAU setting. Thus, several mounted file systems can be active on a server, each with a different DAU setting. The DAU setting is determined when the file system is created using the `sammkfs`(1M) command. It cannot be changed dynamically.

The possible DAU settings differ depending on the file system you are using. The following sections describe the DAU settings for each file system. These sections also introduce the concept of the master configuration (`mcf`) file. You create this ASCII file at system configuration time. It defines the devices and file systems used in your StorageTek QFS or StorageTek ASM environment. The `mcf` file is introduced in the following sections, but it is more thoroughly discussed in "Volume Management" on page 35.

Two file allocation schemes are available to you: a dual allocation scheme and a single allocation scheme. The following sections describe these schemes.

Dual Allocation Scheme

A StorageTek ASM file system is defined as Equipment Type `ms` in your `mcf` file. The only device type allowed in a StorageTek ASM file system is type `md`. Both metadata and file data are written to the `md` devices in a StorageTek ASM file system. By default, the DAU on an `md` device is 16 kilobytes.

A StorageTek QFS file system is defined as Equipment Type `ma` in your `mcf` file. Metadata is written to `mm` devices. Data can be written to `md`, `mr`, or `g`*XXX* devices.

The `md` and `mm` devices use a dual allocation scheme and are as follows:

- On `md` data devices, the small allocation is 4 kilobytes, and the large allocation is a DAU. The default DAU is 64 kilobytes in a StorageTek QFS shared file system. The default DAU is 16 kilobytes in a StorageTek QFS (unshared) file system. You can override this default when the file system is initialized by using the -`a` *allocation_unit* option to the `sammkfs`(1M) command. The DAU size can be either 16, 32, or 64 kilobytes.

When a file is created on an `md` device, the system allocates the first eight addresses of a file in the small allocation. If more space is needed, the file system uses one or more large allocations (DAUs) in expanding the file. As a result, I/O performance improves for large files while minimizing the disk fragmentation that can result from having many small files.

- On `mm` metadata devices, the small allocation is 4 kilobytes, and the large allocation is 16 kilobytes. The dual allocation scheme enables the file system to write metadata to disk more efficiently and helps minimize disk fragmentation.

Depending on the type of file data stored in the file system, selecting a larger DAU size can improve file system performance significantly. For information about tuning file system performance, see "Advanced Topics" on page 281.

Single Allocation Scheme

Only StorageTek QFS file systems can include devices that use a single allocation scheme. The StorageTek QFS file systems are Equipment Type `ma` in your `mcf` file. These file systems consist of separate metadata devices and data devices, as follows:

- The metadata devices can be defined only as Equipment Type `mm`.
- The data devices can be defined as Equipment Type `md`, `mr`, or `g`*XXX*. The `md` devices follow the dual allocation scheme of a StorageTek ASM file system and are limited to DAU sizes of 16 kilobytes, 32 kilobytes, or 64 kilobytes.

  The `mr` and `g`*XXX* devices follow a single allocation scheme. You can mix `mr` and `g`*XXX* devices in a file system, but you cannot mix `md` devices with either `mr` or `g`*XXX* devices in a file system.

  The DAU size for StorageTek QFS file systems that use `mr` and `g`*XXX* data devices is configurable. The possible DAU sizes that can be used on data devices depend on the Equipment Type assigned to each data device in the `mcf` file. Table 7. shows these DAU sizes.fc

Table 7. StorageTek QFS Equipment Types

Equipment Type	DAU Sizes
`mr` or `g`*XXX*	You can specify different DAU sizes by adjusting the default size in 8-kilobyte increments. The DAU size can be from 16 kilobytes to 65,528 kilobytes (64 megabytes). The default DAU for an `mr` or `g`*XXX* device in a StorageTek QFS environment is 64 kilobytes.
`md`	This type of device uses a dual allocation in the style of a StorageTek ASM file system. The DAU can be configured to be 16, 32, or 64 kilobytes in length. The default DAU for an `md` device in a StorageTek QFS environment is 64 kilobytes. An `md` device in a StorageTek QFS file system is used to store data only, not metadata. This is the difference between an `md` device in a StorageTek QFS file system and an `md` device in a StorageTek ASM file system.

Note: If you did not perform a `sammkfs`(1M) on your file system when the StorageTek QFS software was installed, you are using a version 1 superblock. In the version 1 superblock, the `mm` devices do not use the dual allocation scheme. In the version 1 superblock, the allocation for `mm` devices is 16 kilobytes. Only a version 2 superblock enables you to define `md` devices in a StorageTek QFS file system.

The DAU setting is specified using the `-a` *allocation_unit* option to the `sammkfs`(1M) command. The following command specifies a DAU of 128 kilobytes:

```
# sammkfs -a 128 samqfs1
```

For more information about the `sammkfs`(1M) command, see the `sammkfs`(1M) man page.

Allocation Scheme Summary

Table 8. shows the Equipment Types that can be used in StorageTek QFS and StorageTek ASM file systems.

Table 8. Equipment Types for File System Devices

Equipment Types in `mcf` File	Type of Data Stored	Allocation Scheme	File Systems That Can Include the Equipment Type
`md`	File data and metadata	Dual	StorageTek ASM
`md`	File data	Dual	StorageTek QFS
`mm`	Metadata	Dual	StorageTek QFS
`mr`	File data	Single	StorageTek QFS
`g`*XXX*	File data	Single	StorageTek QFS

Within a StorageTek ASM file system (an `ms` file system), you can have only `md` devices.

Within a StorageTek QFS file system (an `ma` file system), you can mix devices as follows:

- `mm` and `mr` devices
- `mm` and `g`*XXX* devices
- `mm`, `mr`, and `g`*XXX* devices
- `mm` and `md` devices

Table 9. summarizes the allocation schemes used by the various file systems.

Table 9. File Allocation

File System and Device Type	Allocation Increments
StorageTek ASM with `md` devices	Up to eight 4-kilobyte blocks, then DAUs
StorageTek QFS with `mr` devices	DAUs
StorageTek QFS with `g`*X* devices	DAUs
StorageTek QFS with `md` devices	Up to eight 4-kilobyte blocks, then DAUs

Table 10. summarizes the DAU defaults.

Table 10. Default DAU Sizes

File System and Device Types	Default DAU Size
StorageTek ASM `md` devices	16 kilobytes
StorageTek QFS `mr` and `md` devices	64 kilobytes
StorageTek QFS `g`*X* devices	256 kilobytes

Stripe Widths on Data Disks

Stripe width defaults differ between StorageTek QFS and StorageTek ASM file systems. The stripe width is specified by the `-o stripe=`*n* option on the `mount`(1M) command. If the stripe width is set to 0, round-robin allocation is used.

The following sections explain the differences that affect stripe widths on the various file systems.

StorageTek ASM Stripe Widths

On StorageTek ASM file systems, the stripe width is set at mount time. Table 11. shows default stripe widths.

Table 11. StorageTek ASM Default Stripe Widths

DAU	Default Stripe Width	Amount of Data Written to 1 Disk
16 kilobytes (default)	8 DAUs	128 kilobytes
32 kilobytes	4 DAUs	128 kilobytes
64 kilobytes	2 DAUs	128 kilobytes

For example, if `sammkfs`(1M) is run with default settings, the default large DAU is 16 kilobytes. If no stripe width is specified when the `mount`(1M) command is issued, the default is used, and the stripe width set at mount time is 8.

Note that if you multiply the number in the first column of Table 11. by the number in the second column, the resulting number is 128 kilobytes. The StorageTek QFS and StorageTek ASM file systems operate more efficiently if the amount of data being written to disk is at least 128 kilobytes.

StorageTek QFS Stripe Widths – Not Using Striped Groups

On StorageTek QFS file systems, the stripe width that is set at mount time depends on whether or not striped groups are configured. A *striped group* is a collection of devices that are striped as a group. For more information about striped groups, see "File Allocation Methods" on page 23. This section describes stripe widths for StorageTek QFS file systems that are configured without stripe groups.

If striped groups are not configured, the DAU and stripe width relationships are similar to those for StorageTek ASM file systems. The differences being that DAUs larger than 64 kilobytes or greater are possible and that the DAU is configurable in 8-kilobyte blocks. The maximum DAU size is 65528 kilobytes.

By default, if no stripe width is specified, the amount of data written to disk is at or near 128 kilobytes. The StorageTek QFS file systems are more efficient if write operations write at least one whole stripe per I/O request. Table 12. shows the default stripe widths. These are the widths used if you do not specify a stripe width.

Table 12. Default Stripe Widths

DAU	Default Stripe Width	Amount of Data Written to 1 Disk
16 kilobytes	8 DAUs	128 kilobytes
24 kilobytes	5 DAUs	120 kilobytes
32 kilobytes	4 DAUs	128 kilobytes
40 kilobytes	3 DAUs	120 kilobytes
48 kilobytes	2 DAUs	96 kilobytes
56 kilobytes	2 DAUs	112 kilobytes
64 kilobytes (default)	2 DAUs	128 kilobytes
72 kilobytes	1 DAU	72 kilobytes
128 kilobytes	1 DAU	128 kilobytes
> 128 kilobytes	1 DAU	DAU size

StorageTek QFS Stripe Widths – Using Striped Groups

If striped groups are configured for your StorageTek QFS file system, the minimum amount of space allocated is the DAU multiplied by the number of devices in the striped group. The amount of the allocation can be very large when using striped groups.

When striped groups are used, data is written to several disk devices at once. This allocation treats a group of disks as if they were one device. Allocations on striped groups are logically equal to the DAU size multiplied by the number of elements in the striped group.

The `-o stripe=`*n* mount option determines the number of allocations that occur on each stripe group before the allocation moves to a different striped group. If a file system is mounted with `-o stripe=0`, the allocation is always to one striped group.

By default, the setting is `-o stripe=0`, which is round robin. The setting can be as low as `-o stripe=0` (which disables striping) or as high as `-o stripe=255`. The system sets `-o stripe=0` if mismatched striped groups are present. When mismatched striped groups are present, a file can reside on only one stripe group.

StorageTek QFS Data Alignment

Data alignment refers to matching the allocation unit of the RAID controller with the allocation unit of the file system. The optimal StorageTek QFS file system alignment formula is as follows:

allocation_unit = *RAID_stripe_width* x *number_of_data_disks_in_the_RAID*

For example, if a RAID-5 unit has a total of nine disks, with one of the nine being the parity disk, the number of data disks is eight. If the RAID stripe width is 64 kilobytes, then the optimal allocation unit is 64 multiplied by 8, which is 512 kilobytes.

Data files are striped or round-robin through each striped group (`g`*XXX*) or data disk (`mr` or `md`) defined within the same file system.

A mismatched alignment hurts performance because it can cause a read-modify-write operation. The rest of this chapter provides more information for you to consider when setting DAUs and determining stripe widths.

Stripe Widths on Metadata Disks

You can use the `-o mm_stripe=`*n* option to the `mount_samfs`(1M) command to stripe metadata information about the metadata disk. The default stripe width is `-o mm_stripe=1`, which specifies that one 16-kilobyte DAU be written to a metadata disk before switching to the next metadata disk. The small 4-kilobyte DAU is used for metadata disks.

By default, if you have multiple metadata devices, metadata is allocated using striped or round-robin allocation depending what is specified on the `-o mm_stripe=`*n* option to the `mount`(1M) command. The setting can be as low as `-o mm_stripe=0`, which disables striping. It can also be as high as `-o mm_stripe=255`.

File Allocation Methods

The StorageTek QFS and StorageTek ASM file systems enable you to specify both round-robin and striped allocation methods. Table 13. shows the default file allocation methods used.

Table 13. Default Allocation Methods

File System	Metadata	File Data
StorageTek ASM	Striped	Striped
StorageTek QFS	Striped	Striped
StorageTek QFS (striped groups)	Striped	Round-robin
StorageTek QFS shared file system	Striped	Round-robin

The following sections describe allocation in more detail.

Metadata Allocation

Metadata allocation differs depending on the type of file system you have.

- For StorageTek ASM file systems, metadata is allocated across the `md` devices.
- For StorageTek QFS file systems, metadata is allocated across the `mm` devices.

No file data is allocated on the `mm` devices in StorageTek QFS file systems.

Inodes are 512 bytes in length. Directories are initially 4 kilobytes in length. Table 14. shows how the system allocates metadata.

Table 14. Metadata Allocation

Metadata Type	Allocation Increments for StorageTek QFS File Systems	Allocation Increments for StorageTek ASM File Systems
Inodes (`.inodes` file)	16-kilobyte DAU	16, 32, or 64-kilobyte (DAU)
Indirect blocks	16-kilobyte DAU	16, 32, or 64-kilobyte (DAU)
Directories	4-kilobyte blocks and 16-kilobyte DAUs	4 kilobytes, up to a 32-kilobyte total, then DAU size

Round-Robin Allocation

The round-robin allocation method writes one data file at a time to each successive device in the family set. Round-robin allocation is useful for multiple data streams because aggregate performance can exceed striping performance in this type of environment.

Round-robin disk allocation enables a single file to be written to a logical disk. The next file is written to the next logical disk. When the number of files written equals the number of devices defined in the family set, the file system starts over again with the first devices selected. If a file exceeds the size of the physical device, the first portion of the file is written to the first device, and the remainder of the file is written to the next device with available storage.

The size of the file being written determines the I/O size. You can specify round-robin allocation explicitly in the `/etc/vfstab` file by entering `stripe=0`.

The following figures depict round-robin allocations. In these figures, file 1 is written to disk 1, file 2 is written to disk 2, file 3 is written to disk 3, and so on. When file 6 is created, it is written to disk 1, starting the round-robin allocation scheme over again.

Figure 2. depicts a StorageTek ASM file system using round-robin allocation on five devices. Figure 3. depicts a StorageTek QFS file system using round-robin allocation on five devices.

Figure 2. Round-Robin StorageTek ASM File System Using Five Devices

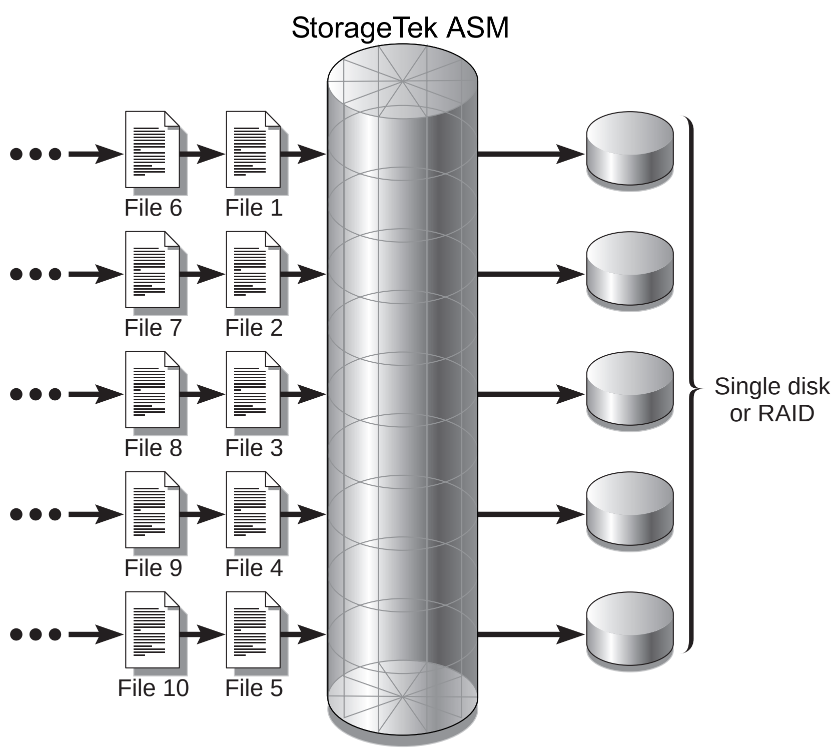
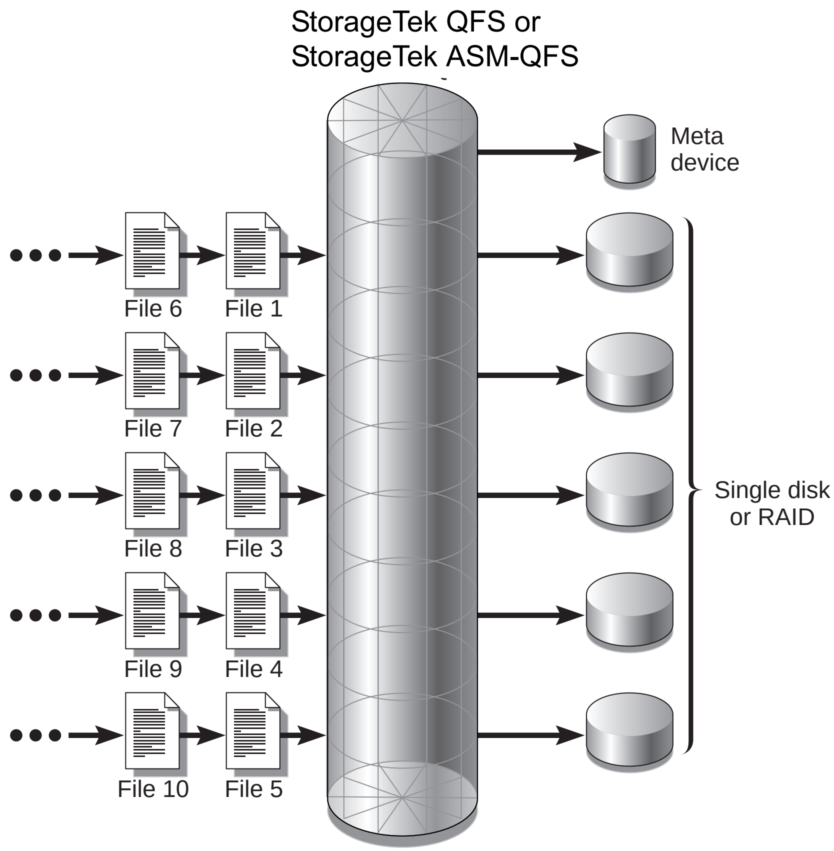


Figure 3. Round-Robin StorageTek QFS File System Using Five Devices



Striped Allocation

By default, StorageTek QFS and StorageTek ASM file systems use a striped allocation method to spread data over all the devices in the file system family set. Striping is a method of writing files in an interlaced fashion across multiple devices concurrently.

Striping is used when performance for one file requires the additive performance of all the devices. A file system that is using striped devices addresses blocks in an interlaced fashion rather than sequentially. Striping generally increases performance because disk reads and writes are spread concurrently across disk heads. Striped disk access enables multiple I/O streams to simultaneously write a file across multiple disks. The DAU and the stripe width determine the size of the I/O transmission.

In a file system using striping, file 1 is written to disk 1, disk 2, disk 3, disk 4, and disk 5. File 2 is written to disks 1 through 5 as well. The DAU multiplied by the stripe width determines the amount of data written to each disk in a block.

When a StorageTek QFS or StorageTek ASM file system starts to write a file to an `md` device, it first assumes that the file will fit into a small DAU, which is 4 kilobytes. If the file does not fit into the first eight small DAUs (32 kilobytes) allocated, the file system writes the remainder of the file into one or more large DAUs.

When a StorageTek QFS file system starts to write a file to an `mr` device, it writes first to one DAU, then another, and so on. The `mr` devices have only one DAU size. A StorageTek QFS file system can also write metadata to striped `mm` devices.

Multiple active files cause significantly more disk head movement if striped allocation is used. If I/O is to occur to multiple files simultaneously, use round-robin allocation.

The following figures depict file systems using striped allocations. In these figures, DAU x *stripe_width* bytes of the file are written to disk 1. DAU x *stripe_width* bytes of the file are written to disk 2. DAU x *stripe_width* bytes of the file are written to disk 3, and so on. The order of the stripe is first-in-first-out for the files. Striping spreads the I/O load over all the disks.

Figure 4. depicts a StorageTek ASM file system using five striped devices. Figure 5. depicts a StorageTek QFS file system using five striped devices.

Figure 4. StorageTek ASM File System Using Five Striped Devices

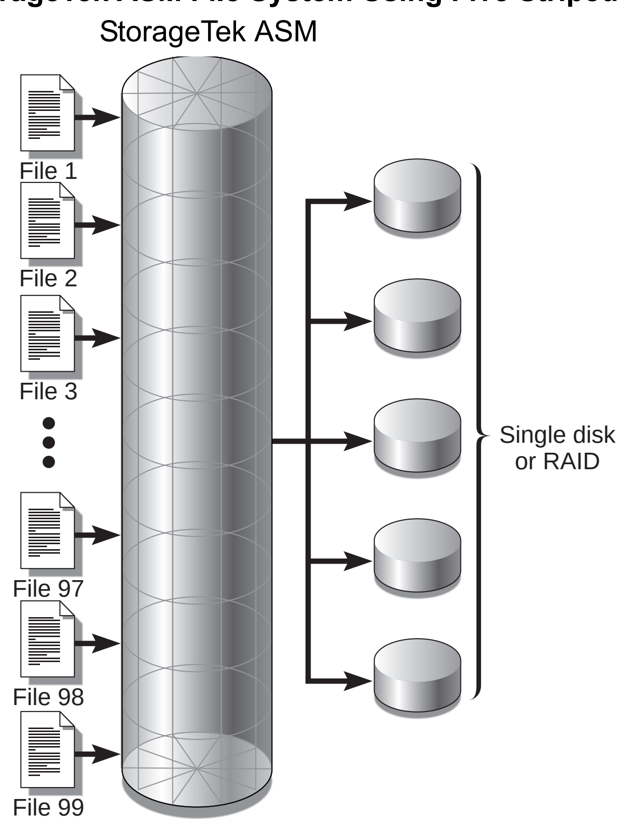
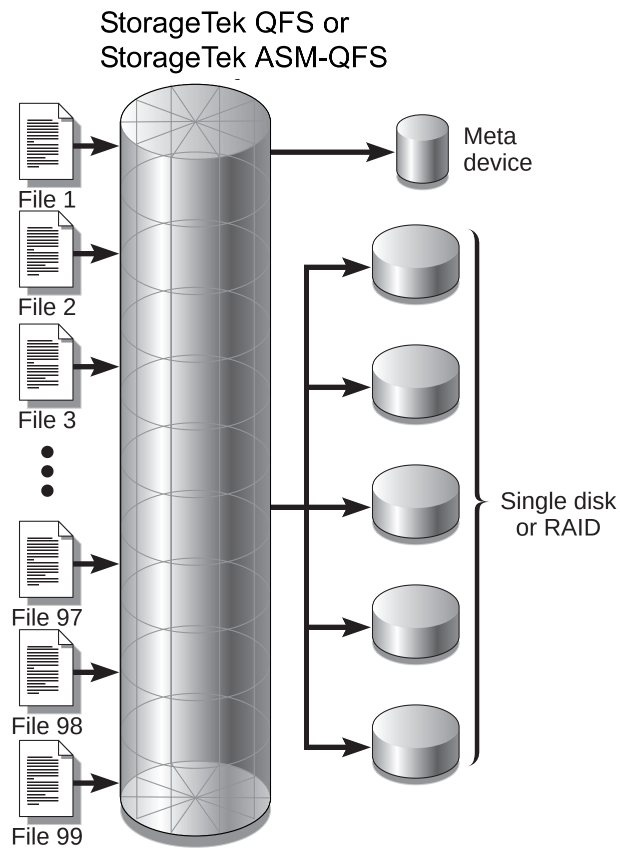


Figure 5. StorageTek QFS File System Using Five Striped Devices



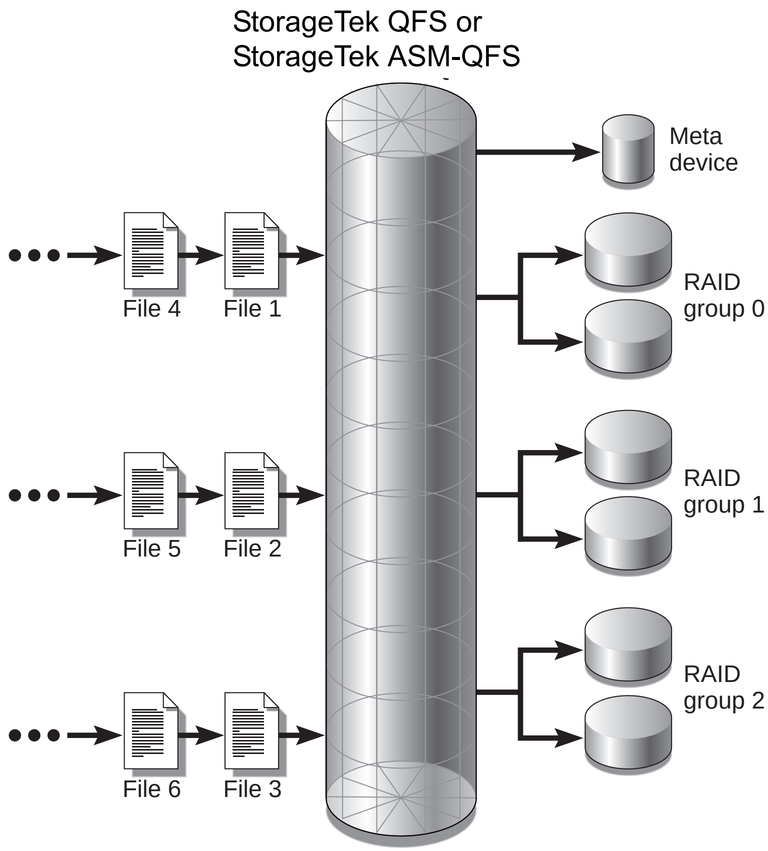
Striped Groups (StorageTek QFS File Systems Only)

A *striped group* is a special StorageTek QFS allocation method designed for file systems that have extremely large I/O requirements and terabytes of disk cache. A striped group enables you to designate an Equipment Type that contains multiple physical disks. Multiple striped group Equipment Types can make up a single StorageTek QFS file system. Striped groups save bit map space and system update time for very large RAID configurations.

A striped group is a collection of devices within a StorageTek QFS file system. Striped groups must be defined in the `mcf` file as `g`*XXX* devices. Striped groups enable one file to be written to and read from two or more devices. You can specify up to 128 striped groups within a file system.

Figure 6. depicts a StorageTek QFS file system using striped groups and a round-robin allocation. In Figure 6., files written to the `qfs1` file system are round-robin between groups `g0`, `g1`, and `g2`. Three striped groups are defined (`g0`, `g1`, and `g2`). Each group consists of two physical RAID devices.

Figure 6. StorageTek QFS Round-Robin Striped Groups



For the configuration in Figure 6., the mount point option in `/etc/vfstab` is set to `stripe=0`. Figure 7. shows the `mcf` file that declares these striped groups.

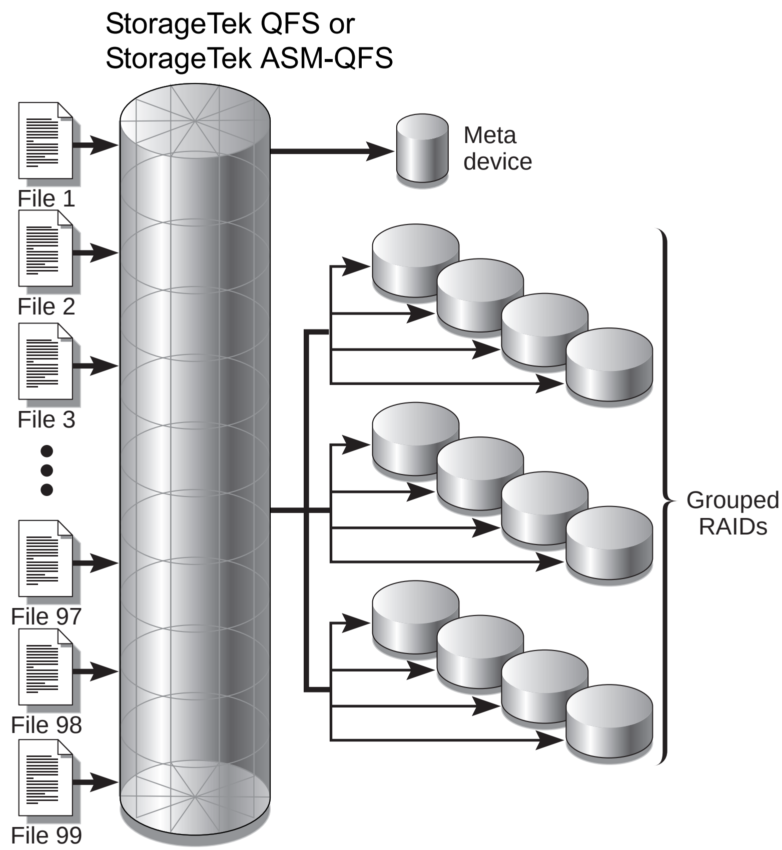
Figure 7. Example `mcf` File Showing Striped Groups

```
# Equipment          Eq   Eq    Fam   Dev    Additional
# Identifier         Ord  Type  Set   State  Parameters
#
qfs1                 10   ma    qfs1
/dev/dsk/c0t1d0s6    11   mm    qfs1  -
/dev/dsk/c1t1d0s2    12   g0    qfs1  -
/dev/dsk/c2t1d0s2    13   g0    qfs1  -
/dev/dsk/c3t1d0s2    14   g1    qfs1  -
/dev/dsk/c4t1d0s2    15   g1    qfs1  -
/dev/dsk/c5t1d0s2    16   g2    qfs1  -
/dev/dsk/c6t1d0s2    17   g2    qfs1  -
```

Figure 8. depicts a StorageTek QFS file system using striped groups in which the data is striped across groups. In Figure 8., files written to the `qfs1` file system are striped through groups `g0`, `g1`, and `g2`. Each group includes four

physical RAID devices. The mount point option in `/etc/vfstab` is set to `stripe=1` or greater.

Figure 8. StorageTek QFS Striped Group Allocation



Mismatched Striped Groups (StorageTek QFS File Systems Only)

It is possible to build a file system with mismatched striped groups. File systems with mismatched striped groups are those that contain multiple striped groups with different numbers of devices in each group. StorageTek QFS file systems support mismatched striped groups, but they do not support striping on mismatched groups. File systems with mismatched striped groups are mounted as round-robin file systems.

Note: If a file system contains mismatched striped groups, a single file can never span more than one stripe group. If the stripe group on which the file resides fills, it cannot be extended. If mismatched stripe groups are present, use the `setfa`(1) command's `-g` option to direct files into the desired group. For more information, see the `setfa`(1) man page.

To determine how full a stripe group is, use the `samu`(1M) operator utility, and access the `m` display to display the status of mass storage.

The following example shows how a file system can be set up to store different types of files.

Example

Assume that you have a StorageTek QFS license, and you need to create a file system at your site that contains both video and audio data.

Video files are quite large and require greater performance than audio files. You want to store them in a file system with a large striped group because striped groups maximize performance for very large files.

Audio files are smaller and require lower performance than video files. You want to store them in a small striped group. One file system can support both video and audio files.

Figure 9. depicts the file system needed. It is a StorageTek QFS file system using mismatched striped groups in a striped allocation

Figure 9. StorageTek QFS File System Using Mismatched Striped Groups in a Striped Allocation .

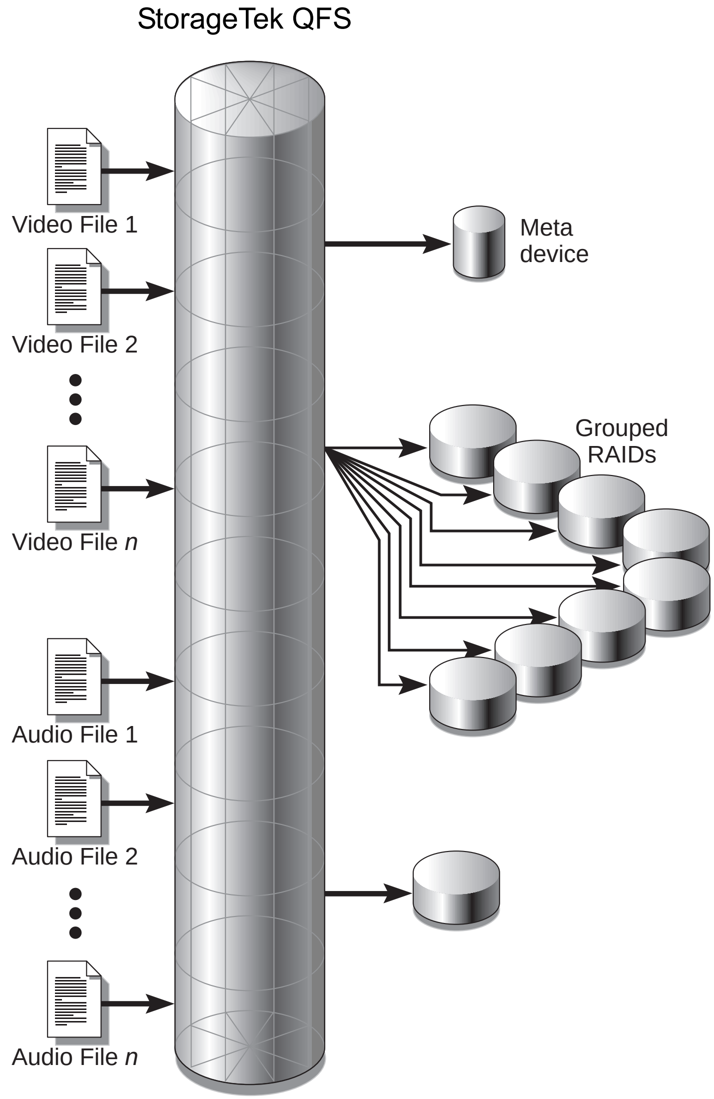


Table 15. shows the characteristics of this file system.

Table 15. File System `avfs` Characteristics

Characteristics	Notes
File system name	`avfs`.
Number of stripe groups	Two. The video file group is `g0`. The audio file group is `g1`.
Stripe width	0.
DAU	128 kilobytes.
Number of disks for `g0`	Eight.

Table 15. File System `avfs` Characteristics (Continued)

Characteristics	Notes
Minimum block size for `g0`	Eight disks x 128-kilobyte DAU = 1024 kilobytes. This is the amount of data written in one block write. Each disk receives 128 kilobytes of data, so the total amount written to all disks at one time is 1024 kilobytes.
Number of disks for `g1`	One.
Minimum block size for `g1`	One disk x 128-kilobyte DAU = 128 kilobytes.

Add the following line to the `/etc/vfstab` file so the environment recognizes the `avfs` file system:

```
avfs    -    /avfs    samfs    -    no    stripe=0
```

Note that in the `/etc/vfstab` file, `stripe=0` is used to specify a round-robin file system. This is used because a value greater than 0 (stripe > 0) is not supported for mismatched striped groups.

Figure 10. shows the `mcf` file for file system `avfs`.

Figure 10. The `mcf` File for File System `avfs`

```
# Equipment          Eq   Eq    Fam   Dev     Additional
# Identifier         Ord  Type  Set   State   Parameters
#
avfs                 100  ma    avfs
/dev/dsk/c00t1d0s6   101  mm    avfs   -
#
/dev/dsk/c01t0d0s6   102  g0    avfs   -
/dev/dsk/c02t0d0s6   103  g0    avfs   -
/dev/dsk/c03t0d0s6   104  g0    avfs   -
/dev/dsk/c04t0d0s6   105  g0    avfs   -
/dev/dsk/c05t0d0s6   106  g0    avfs   -
/dev/dsk/c06t0d0s6   107  g0    avfs   -
/dev/dsk/c07t0d0s6   108  g0    avfs   -
/dev/dsk/c08t0d0s6   109  g0    avfs   -
#
/dev/dsk/c09t1d0s6   110  g1    avfs   -
```

After the `mcf` file for this file system is ready, you can enter the `sammkfs`(1M) and `mount`(1M) commands shown in Figure 11. to create and mount the `avfs` file system.

Figure 11. Commands to Create and Mount File System `avfs`

```
# sammkfs -a 128 avfs
# mount avfs
```

After the file system is mounted, you can use the commands shown in Figure 12. to create two directories for the two types of files.

Figure 12. Commands to Create Directories in File System `avfs`

```
# cd /avfs
# mkdir video
# mkdir audio
```

After the directories are created, you can use the `setfa`(1) commands shown in Figure 13. to assign the large striped group to video and to assign the small striped group to audio. Files created in these directories are allocated on their respective striped groups because attributes are inherited.

Figure 13. Commands to Set File Attributes

```
# setfa -g0 video
# setfa -g1 audio
```

For more information about the `sammkfs`(1M) command, see the `sammkfs`(1M) man page. For more information about the `mount`(1M) commands, see the `mount_samfs`(1M) man page. For more information about the `setfa`(1) command, see the `setfa`(1) man page.

Volume Management

3

The master configuration file (`mcf`) describes all devices that are under the control of, or used by, the StorageTek QFS or StorageTek ASM software. When you create this file, you declare attributes for each device, and you group the devices comprising each file system into family sets.

The installation and configuration process is described completely in the *StorageTek ASM Installation and Configuration Guide*. This chapter provides additional information about configuring the file systems used in the StorageTek QFS and StorageTek ASM environments. This chapter contains the following sections:

Note: References to StorageTek ASM also apply to StorageTek ASM-QFS configurations when talking about storage and archive management. References to StorageTek QFS also apply to StorageTek ASM-QFS configurations when talking about file system design and capabilities. This section refers to *StorageTek ASM-QFS* only when needed for clarity.

Creating the mcf File

The first step toward configuring a StorageTek QFS or StorageTek ASM file system is to create a master configuration file in `/etc/opt/SUNWsamfs/mcf`. The `mcf` file contains the information that these file systems need in order to identify and organize RAID and disk devices into file systems. It also contains entries for each automated library or device included in a file system. A sample `mcf` file is located in `/opt/SUNWsamfs/examples/mcf`.

An `mcf` file is an ASCII file that consists of lines of specification code divided into six columns, or fields. Figure 14. shows the six fields in an `mcf` file line.

Figure 14. Fields in an `mcf` File

```
Equipment    Equipment   Equipment   Family   Device   Additional
Identifier   Ordinal     Type        Set      State    Parameters
```

The following rules pertain to how data can be entered in the `mcf` file:

- Enter either space or tab characters between the fields in the file.
- You can include comment lines in an `mcf` file. Comment lines start with a pound character (#).
- Some fields do not need to contain useful information. Use a dash character (-) to indicate that an optional field contains no meaningful information.

For more information about writing the `mcf` file, see the `mcf`(4) man page. You can also use the ASM QFS Manager to create an `mcf` file. For information about installing ASM QFS Manager, see *StorageTek ASM Installation and Configuration Guide*. For information about using ASM QFS Manager, see its online help.

The following sections describe each field in an `mcf` file:

- "The Equipment Identifier Field" on page 36
- "The Equipment Ordinal Field" on page 37
- "The Equipment Type Field" on page 37
- "The Family Set Field" on page 38
- "The Device State Field" on page 39
- "The Additional Parameters Field" on page 39

The Equipment Identifier Field

The Equipment Identifier field is a required field. Use the Equipment Identifier field to specify the following kinds of information:

- The file system name. If this field contains a file system name, it must be identical to the Family Set name, and the subsequent lines in the `mcf` file must define all the disks or devices included in the file system. More than one file system can be declared in an `mcf` file. Typically, the first data line in an `mcf` file declares the first file system, and subsequent lines specify the devices included in the file system. Other file systems declared in the `mcf` file can be preceded by a blank comment line for readability. File system names must start with an alphabetic character and can contain only alphabetic characters, numeric characters, or underscore (_) characters.
- The `nodev` keyword. If this field contains the keyword `nodev`, the `mcf` file is being used as a client host in a StorageTek QFS shared file system on a Solaris host. Do not use this keyword if you are running in a Sun Cluster environment. This keyword can appear in this field only as the Equipment Identifier for one or more metadata devices that reside on the metadata server. For more information about creating an `mcf` file for the members

of a StorageTek QFS shared file system, see the *StorageTek ASM Installation and Configuration Guide*.

- A disk partition or slice description. A `/dev/` entry in this field identifies a disk partition or slice.
- An automated library or optical drive description. If this field is a `/dev/samst` entry, it identifies an automated library or optical drive. If you are configuring a network-attached automated library, see the *StorageTek ASM Installation and Configuration Guide* and the *StorageTek ASM Storage and Archive Manager Guide* for more information.
- A tape drive description. If the field is a tape drive, the entry can be in one of two forms:
  - The field can contain a `/dev/rmt` entry.
  - The field can contain a path to a symbolic link that points to the same special file that the `/dev/rmt` link points to. If you specify a tape drive in this manner, make sure you create the link before mounting the file system.

If the Equipment Identifier field contains the name of a Family Set, it is limited to 31 characters. For all other content, this field is limited to 127 characters.

The Equipment Ordinal Field

For each row in the `mcf` file, the Equipment Ordinal field must contain a numeric identifier for the file system component or device being defined. Specify a unique integer such that $1 \leq eq\_ord \leq 65534$. This is a required field.

The Equipment Type Field

Enter a 2-, 3-, or 4-character code for the Equipment Type field. This is a required field.

As Table 16. shows, a StorageTek ASM file system can contain either `ms` or `md` in the Equipment Type field.

Table 16. StorageTek ASM Equipment Type Field

Equipment Type Field Content	Meaning
`ms`	Defines a StorageTek ASM file system.
`md`	Defines a striped or round-robin device for storing file data and metadata information.

As Table 17. shows, a StorageTek QFS or StorageTek ASM-QFS file system can contain either `ma`, `md`, `mm`, `mr`, or `g`*XXX* in the Equipment Type field.

Table 17. StorageTek QFS or StorageTek ASM-QFS Equipment Type Field

Equipment Type Field Content	Meaning
`ma`	Defines a StorageTek QFS or StorageTek ASM-QFS file system.
`md`	Defines a striped or round-robin device for storing file data.
`mm`	Defines a metadata device for storing inode and other nondata information.
`mr`	Defines a round-robin or striped data device.
`g`*XXX*	Striped group data device. Striped groups start with the letter `g` followed by a number. The number must be an integer such that `0` ≤ *XXX* ≤ `127`. For example, `g12`. All members in a striped group must be the same type and size. Different striped groups within one file system are not required to have the same number of members. `md`, `mr`, and `g`*XXX* devices cannot be mixed in one file system.

Besides the file system equipment types, other codes are used to identify automated libraries and other devices. For more information about specific equipment types, see the `mcf`(4) man page.

The Family Set Field

The Family Set field contains the name for a group of devices. This is a required field.

Family Set names must start with an alphabetic character and can contain only alphabetic characters, numeric characters, or underscore (_) characters.

For lines that define a file system, the lines that define the disk devices in a file system must all contain the same Family Set name. The software uses the Family Set to groups devices with the same Family Set name together as a file system. It physically records the Family Set name on all the devices in the file system when the `sammkfs`(1M) command is issued. You can change this name by using the `-F` and `-R` options together on the `samfsck`(1M) command. For more information about the `sammkfs`(1M) command, see the `sammkfs`(1M) man page. For more information about the `samfsck`(1M) command, see the `samfsck`(1M) man page.

For lines that define an automated library and its associated drives, the lines defining the devices must contain the same Family Set name.

For a standalone, manually loaded removable media device, this field can contain a dash (-).

The Device State Field

The Device State field specifies the state of the device when the file system is initialized. Valid device states are `on` and `off`. This is an optional field. If you do not want to enter `on` or `off`, enter a dash (`-`) character to indicate that this field is omitted.

The Additional Parameters Field

For a StorageTek ASM file system, the Additional Parameters field is optional and can be left completely blank. By default, library catalog files are written to `/var/opt/SUNWsamfs/catalog/`*family_set_name*. Use this field if you want to specifiy an alternate path to the library catalog file.

For a StorageTek QFS shared file system, this field must contain the keyword `shared`.

For a StorageTek QFS unshared file system, enter a dash or leave this field blank.

Examples of mcf Files

Each file system configuration is unique. System requirements and actual hardware differ from site to site. The following sections show sample `mcf` files.

StorageTek ASM Volume Management Example

For the StorageTek ASM file system, you can define family sets in the `/etc/opt/SUNWsamfs/mcf` file in the Equipment Type field using the following equipment types:

- `ms` for the StorageTek ASM file system type.
- `md` for the devices. Data is striped or round-robin across these devices. The stripe width is set with the `-o stripe=`*n* option on the `mount`(1M) command. The default stripe width is set based on the DAU size. For more information about stripe widths and DAU sizes, see “File System Design” on page 7.

Both metadata (including inodes, directories, allocation maps, and so on) and file data on StorageTek ASM file systems are located on the same disk. Data files are striped or round-robin through each disk partition defined within the same file system.

Figure 15. shows an `mcf` file for a StorageTek ASM file system.

Figure 15. Example `mcf` File for a StorageTek ASM File System

```
# StorageTek ASM file system configuration example
#
# Equipment        Eq    Eq    Fam.   Dev.     Additional
# Identifier       Ord   Type  Set    State    Parameters
#-----------       ---   --   ------  ------   ------------------
samfs1              10   ms   samfs1
/dev/dsk/c1t1d0s6   11   md   samfs1   -
/dev/dsk/c2t1d0s6   12   md   samfs1   -
/dev/dsk/c3t1d0s6   13   md   samfs1   -
/dev/dsk/c4t1d0s6   14   md   samfs1   -
/dev/dsk/c5t1d0s6   15   md   samfs1   -
```

StorageTek QFS and StorageTek ASM-QFS Volume Management Examples

For the StorageTek QFS and StorageTek ASM-QFS file systems, family sets are defined in the `/etc/opt/SUNWsamfs/mcf` file in the Equipment Type field using the following equipment types:

- `ma` for the StorageTek QFS or StorageTek ASM-QFS file system type.
- `mm` for a metadata device. File data is not written to this device. You can specify multiple metadata devices. Metadata (including inodes, directories, allocation maps, and so on) on StorageTek QFS and StorageTek ASM-QFS file systems is located on the metadata device(s) and is separated from the file data devices. By default, metadata is allocated using round-robin allocation if you have multiple metadata devices.
- `mr` or `md` for devices upon which file data is to be striped or round-robin.
- `g`*XXX* for devices upon which file data is to be striped as a group. A striped group is a logical group of devices that are striped as a unit. Data is striped across the members of each group.

  Groups are specified with `g0` through `g127` equipment type numbers, with the stripe width on each device being the DAU. All devices in a striped group must be the same size. Different striped groups within one file system are not required to have the same number of members. `mr` and `g`*XXX* devices can be mixed in a file system, but `md` devices cannot be mixed with either `mr` or `g`*XXX* devices in a file system.

  Data can be striped (if all groups contain the same number of devices) or round-robin between groups. The default is round robin.

Data files are striped or round-robin through each data disk partition (`mr` or `g`*XXX*) defined within the same file system.

Example 1

Figure 16. shows an `mcf` file for a StorageTek QFS or StorageTek ASM-QFS file system with two striped groups.

Figure 16. Example `mcf` File Showing Striped Groups

```
# StorageTek QFS file system configuration
#
# Equipment        Eq    Eq    Fam.   Dev.    Additional
# Identifier       Ord   Type  Set    State   Parameters
#-----------       ---    --  ------ ------   ------------------
qfs1                10    ma   qfs1     -
/dev/dsk/c2t1d0s7   11    mm   qfs1     -
/dev/dsk/c3t0d0s6   12    g0   qfs1     -
/dev/dsk/c3t0d1s6   13    g0   qfs1     -
/dev/dsk/c4t0d0s6   14    g1   qfs1     -
/dev/dsk/c4t0d1s6   15    g1   qfs1     -
```

Example 2

Figure 17. shows an `mcf` file with three StorageTek ASM-QFS file systems.

Figure 17. Example `mcf` File Showing Three StorageTek ASM-QFS File Systems

```
# StorageTek ASM-QFS file system configuration example
#

# Equipment        Eq     Eq   Fam.   Dev.    Additional
# Identifier       Ord   Type  Set    State   Parameters
#-----------       ---    --  ------ ------   ------------------
qfs1                10   ma   qfs1    -
/dev/dsk/c1t13d0s6 11   mm   qfs1    -
/dev/dsk/c1t12d0s6 12   mr   qfs1    -
#
qfs2                20   ma   qfs2    -
/dev/dsk/c1t5d0s6   21   mm   qfs2    -
/dev/dsk/c5t1d0s6   22   mr   qfs2    -
#
qfs3                30   ma   qfs3    -
/dev/dsk/c7t1d0s3   31   mm   qfs3    -
/dev/dsk/c6t1d0s6   32   mr   qfs3    -
/dev/dsk/c6t1d0s3   33   mr   qfs3    -
/dev/dsk/c5t1d0s3   34   mr   qfs3    -
```

Example 3

Figure 18. shows an `mcf` file with one StorageTek ASM-QFS file system that uses `md` devices. This `mcf` file also defines a tape library.

Figure 18. Example `mcf` File Showing a StorageTek ASM-QFS File System and a Library

```
# StorageTek ASM-QFS file system configuration example
#

# Equipment          Eq    Eq   Fam.   Dev.     Additional
# Identifier         Ord  Type  Set    State    Parameters
#-----------         ---   --  ------ ------   ----------
samfs1               10    ma   samfs1  -
/dev/dsk/c1t2d0s6    11    mm   samfs1  -
/dev/dsk/c1t3d0s6    12    md   samfs1  -
/dev/dsk/c1t4d0s6    13    md   samfs1  -
/dev/dsk/c1t5d0s6    14    md   samfs1  -
# scalar 1000 and 12 AIT tape drives
/dev/samst/c5t0u0    30    rb   robot1  -
/dev/rmt/4cbn       101    tp   robot1  on
/dev/rmt/5cbn       102    tp   robot1  on
/dev/rmt/6cbn       103    tp   robot1  on
/dev/rmt/7cbn       104    tp   robot1  off
/dev/rmt/10cbn      105    tp   robot1  on
/dev/rmt/11cbn      106    tp   robot1  on
/dev/rmt/3cbn       107    tp   robot1  on
/dev/rmt/2cbn       108    tp   robot1  on
/dev/rmt/1cbn       109    tp   robot1  on
/dev/rmt/0cbn       110    tp   robot1  on
/dev/rmt/9cbn       111    tp   robot1  on
/dev/rmt/8cbn       112    tp   robot1  on
```

For more examples showing file system configuration in the `mcf` file, see the *StorageTek ASM Installation and Configuration Guide*.

Interactions Between File Settings, Options, and Directives

The `mcf` file defines each file system, but file system behavior depends on interactions between default systems settings, settings in the `/etc/vfstab` file, settings in the `samfs.cmd` file, and options on the `mount`(1M) command line.

You can specify some mount options, for example the stripe width, in more than one place. When this happens, settings in one place can override the settings in another.

For information about the various ways to specify mount options, see “Mounting a File System” on page 61.

Initializing a File System

The `sammkfs`(1M) command constructs new file systems, and its `-a` *allocation_unit* option allows you to specify the DAU setting. The number specified for *allocation_unit* determines the DAU setting.

The 4.2 releases of these file systems support two different superblock designs. Both superblock designs are available to you in the 4.2 release. In Figure 19., the `samfsinfo`(1M) command output shows which superblock a file system is using.

Figure 19. `samfsinfo`**(1M) Command Example**

```
# samfsinfo samfs1
name:     samfs1      version:      2
time:     Wed Feb 21 13:32:18 1996
count:    1
capacity:      001240a0         DAU:         16
space:         000d8ea0
ord  eq   capacity      space    device
  0  10   001240a0   000d8ea0    /dev/dsk/c1t1d0s0
```

The first line of the preceding output indicates that this is a version 2 superblock. Be aware of the following operational and feature differences that pertain to these superblocks:

- Releases prior to 4.0 support only the version 1 superblock design.
- The 4.0 and later releases support the version 2 superblock. If you installed the 4.0 software as an upgrade, you must use the 4.0 or 4.2 `sammkfs`(1M) command to reinitialize your existing file systems before you attempt to use any of the features that depend on the version 2 superblock. Certain features, such as access control lists (ACLs) and the StorageTek QFS shared file system, are supported only in the version 2 superblock. Reinitializing a file system is described as a step in the 4.2 software installation upgrade process, but you can do this any time after the software is installed.

CAUTION: File systems that use a version 2 superblock cannot revert to a release prior to 4.0. You cannot use 4.2 release software to create a version 1 superblock.

For more information about features that require a version 2 superblock, or on using the `sammkfs`(1M) command to create the version 2 superblock, see the *StorageTek ASM Installation and Configuration Guide*.

Example

Figure 20. shows using the `sammkfs`(1M) command to initialize a StorageTek ASM file system using a version 2 superblock.

Figure 20. Initializing a File System with a Version 2 Superblock

```
# sammkfs -a 64 samfs1
Building 'samfs1' will destroy the contents of devices:
                /dev/dsk/c1t9d0s2
                /dev/dsk/c8t1d0s2
                /dev/dsk/c8t5d0s2
                /dev/dsk/c8t6d0s2
Do you wish to continue? [y/N] y
total data kilobytes       = 1715453952
total data kilobytes free  = 1715453760
total meta kilobytes       = 17684128
total meta kilobytes free  = 17680304
```

For more information about the `sammkfs`(1M) command, see the `sammkfs`(1M) man page.

Configuration Examples

The rest of this chapter presents sample configurations and shows various steps and considerations in setting up the `mcf` file on a server. The following procedures are described:

- "To Create a StorageTek QFS Round-Robin Disk Configuration" on page 45
- "To Create a StorageTek ASM Round-Robin Disk Configuration" on page 46
- "To Create a StorageTek QFS Striped Disk Configuration" on page 47
- "To Create a StorageTek ASM Striped Disk Configuration" on page 48
- "To Create a StorageTek QFS Striped Groups Configuration" on page 49

Note that all sample StorageTek QFS configurations could have automated libraries and other removable media devices defined as well, essentially extending the file system beyond the size of the disk cache. Removable media device configurations are shown in only one example. For information about configuring removable media devices see the *StorageTek ASM Installation and Configuration Guide*.

The sample configurations assume that the file system is loaded on the system and all file systems are unmounted.

To Create a StorageTek QFS Round-Robin Disk Configuration

This sample configuration illustrates a StorageTek QFS file system that separates the metadata onto a low-latency disk. Round-robin allocation is used on four partitions. Each disk is on a separate controller.

This procedure assumes the following:

- The metadata device is a single partition (`s6`) used on controller 5, logical unit number (LUN) 0 of the device designated as Equipment Ordinal 11.
- The data devices consist of four disks attached to four controllers.

1. Use an editor to create the `mcf` file.

   Figure 21. shows an example `mcf` file.

Figure 21. Example StorageTek QFS Round Robin `mcf` File

```
# StorageTek QFS disk cache configuration
# Round-robin mcf example

# Equipment          Eq   Eq    Fam.   Dev      Additional
# Identifier         Ord  Type  Set    State    Parameters
#-----------         ---   --  ------ ------   ------------------
qfs1                   1   ma   qfs1
/dev/dsk/c5t0d0s6   11   mm   qfs1    on
/dev/dsk/c1t1d0s6   12   mr   qfs1    on
/dev/dsk/c2t1d0s6   13   mr   qfs1    on
/dev/dsk/c3t1d0s6   14   mr   qfs1    on
/dev/dsk/c4t1d0s6   15   mr   qfs1    on
```

2. Use the `mkdir`(1) command to create the `/qfs` mount point for the `/qfs1` file system.

   For example:

```
# mkdir /qfs
```

3. Use the `sammkfs`(1M) command to initialize the file system.

   The following example uses the default 64-kilobyte DAU:

```
# sammkfs qfs1
```

4. Use an editor to modify the `/etc/vfstab` file.

   The StorageTek QFS file system with `mr` data devices uses striped allocation as a default, so you must set `stripe=0` for round-robin allocation. To explicitly set round-robin on the file system, set the `stripe=0`, as follows:

```
qfs1     -     /qfs     samfs     -     yes     stripe=0
```

5. Use the `mount`(1M) command to mount the file system.

   For example:

```
# mount /qfs
```

To Create a StorageTek ASM Round-Robin Disk Configuration

This sample configuration illustrates a StorageTek ASM file system. Striped allocation is used by default on four partitions. You must set `stripe=0` to specify round-robin allocation. The file system is created using the `sammkfs`(1M) command. The data devices consist of four disks attached to four controllers. Each disk is on a separate controller.

1. Use an editor to create the `mcf` file.

   Figure 22. shows an example `mcf` file.

Figure 22. Example StorageTek ASM Round Robin `mcf` File

```
# StorageTek ASM disk cache configuration
# Round-robin mcf example

# Equipment         Eq    Eq    Fam.    Dev     Additional
# Identifier        Ord   Type  Set     State   Parameters
#-----------        ---   --   ------  ------  -----------------
samfs1               1    ms    samfs1
/dev/dsk/c1t1d0s6   11    md    samfs1   on
/dev/dsk/c2t1d0s6   12    md    samfs1   on
/dev/dsk/c3t1d0s6   13    md    samfs1   on
/dev/dsk/c4t1d0s6   14    md    samfs1   on
```

2. Use the `mkdir`(1) command to create the `/samfs` mount point for the `/samfs1` file system.

   For example:

```
# mkdir /samfs
```

3. Use the `sammkfs`(1M) command to initialize the file system.

The default DAU is 16 kilobytes, but the following example sets the DAU size to 64 kilobytes:

```
# sammkfs -a 64 samfs1
```

4. Use an editor to modify the `/etc/vfstab` file.

   The StorageTek ASM file system uses striped allocation by default, so you must set `stripe=0` for round-robin allocation. To explicitly set round-robin on the file system, set the `stripe=0`, as follows:

```
samfs1    -    /samfs    samfs    -    yes    stripe=0
```

5. Use the `mount`(1M) command to mount the file system.

   For example:

```
# mount /samfs
```

To Create a StorageTek QFS Striped Disk Configuration

This sample configuration illustrates a StorageTek QFS file system. By default, file data is striped to four data partitions.

This procedure assumes the following:

- The metadata device is a single partition (`s6`) used on controller 0, LUN 1. Metadata is written to equipment 11 only.
- The data devices consist of four disks attached to four controllers. Each disk is on a separate controller.

1. Use an editor to create the `mcf` file.

   Figure 23. shows an example `mcf` file.

Figure 23. Example StorageTek QFS Striped Disk `mcf` File

```
# StorageTek QFS disk cache configuration
# Striped Disk mcf example

# Equipment         Eq   Eq    Fam.   Dev.    Additional
# Identifier        Ord  Type  Set    State   Parameters
#-----------        ---   --  ------ ------   ------------------
qfs1                 10   ma  qfs1
/dev/dsk/c0t1d0s6    11   mm  qfs1     on
/dev/dsk/c1t1d0s6    12   mr  qfs1     on
/dev/dsk/c2t1d0s6    13   mr  qfs1     on
/dev/dsk/c3t1d0s6    14   mr  qfs1     on
/dev/dsk/c4t1d0s6    15   mr  qfs1     on
```

2. Use the `mkdir`(1) command to create the `/qfs` mount point for the `/qfs1` file system.

   For example:

```
# mkdir /qfs
```

3. Use the `sammkfs`(1M) command to initialize the file system.

   The default DAU is 64 kilobytes, but the following example sets the DAU size to 128 kilobytes:

```
# sammkfs -a 128 qfs1
```

   With this configuration, any file written to this file system is striped across all of the devices in increments of 128 kilobytes.

4. Use an editor to modify the `/etc/vfstab` file.

   The StorageTek ASM file system uses striped allocation by default. This example sets the stripe width as `stripe=1` DAU, which is the default. The following setting stripes data across all four of the `mr` devices with a stripe width of one DAU:

```
qfs1    -    /qfs    samfs    -    yes    stripe=1
```

5. Use the `mount`(1M) command to mount the file system.

   For example:

```
# mount /qfs
```

To Create a StorageTek ASM Striped Disk Configuration

This sample configuration illustrates a StorageTek ASM file system.

1. Use an editor to create the `mcf` file.

   Figure 24. shows an example `mcf` file. The data devices consist of four disks attached to four controllers. Each disk is on a separate LUN.

Figure 24. Example StorageTek ASM Striped Disk `mcf` File

```
# StorageTek ASM disk cache config
# Striped Disk mcf example

# Equipment        Eq   Eq    Fam.   Dev.    Additional
# Identifier       Ord  Type  Set    State   Parameters
#-----------       ---   --  ------ ------  ------------------
```

Figure 24. Example StorageTek ASM Striped Disk `mcf` File (Continued)

```
samfs1              10   ms  samfs1
/dev/dsk/c1t1d0s6   11   md  samfs1   on
/dev/dsk/c2t1d0s6   12   md  samfs1   on
/dev/dsk/c3t1d0s6   13   md  samfs1   on
/dev/dsk/c4t1d0s6   14   md  samfs1   on
```

2. Use the `mkdir`(1) command to create the `/samfs` mount point for the `/samfs1` file system.

   For example:

```
# mkdir /samfs
```

3. Use the `sammkfs`(1M) command to initialize the file system.

   The following example uses the default 16-kilobyte DAU:

```
# sammkfs samfs1
```

   With this striped disk configuration, any file written to this file system is striped across all of the devices in increments of 16 kilobytes.

4. Use an editor to modify the `/etc/vfstab` file.

   Specify the mount point for this file system in `/etc/vfstab`.

5. Use the `mount`(1M) command to mount the file system.

   For example:

```
# mount /samfs
```

To Create a StorageTek QFS Striped Groups Configuration

Striped groups allow you to group RAID devices together for very large files. A DAU is represented by one bit in the bit maps. If the striped group has *n* devices, *n* multiplied by the DAU is the minimum allocation. Only one bit in the bit maps is used to represent *n* X DAU. This method of writing huge DAUs across RAID devices saves bit map space and system update time. Striped groups are useful for writing very large files to a group of RAID devices and for streaming large amounts of data to and from disk.

Note: The minimum disk space allocated in a striped group is as follows:

minimum_disk_space_allocated = *DAU* x *number_of_disks_in_the_group*

Writing a single byte of data fills the entire minimum disk space allocated in a striped group. Striped groups are used for very specific applications. Make sure that you understand the effects of using striped groups with your file system.

Files with lengths less than the aggregate stripe width times the number of devices (in this example, files less than 128 kilobytes x 4 disks = 512 kilobytes in length) still use 512 kilobytes of disk space. Files larger than 512 kilobytes have space allocated for them as needed in total space increments of 512 kilobytes.

The devices within a striped group must be the same size. It is not possible to add devices to increase the size of a striped group. You can use the `samgrowfs`(1M) command to add additional striped groups, however. For more information about this command, see the `samgrowfs`(1M) man page.

This sample configuration illustrates a StorageTek QFS file system that separates the metadata onto a low-latency disk. Two striped groups are set up on four drives.

This procedure assumes the following:

- The metadata device is a single partition (`s6`) used on controller 0, LUN 1.
- The data devices consist of four disks (two groups of two identical disks) attached to four controllers. Each disk is on a separate LUN. The entire disk is used for data storage, assuming that partition 6 occupies the entire disk.

1. Use an editor to create the `mcf` file.

   Figure 25. shows an example `mcf` file.

Figure 25. Example StorageTek QFS Striped Group `mcf` File

```
# StorageTek QFS disk cache configuration
# Striped Groups mcf example

# Equipment         Eq    Eq     Fam.   Dev.     Additional
# Identifier        Ord   Type   Set    State    Parameters
#-----------       ---    --   ------ ------   ------------------
qfs1                 10    ma   qfs1
/dev/dsk/c0t1d0s6   11    mm   qfs1     on
/dev/dsk/c1t1d0s6   12    g0   qfs1     on
/dev/dsk/c2t1d0s6   13    g0   qfs1     on
/dev/dsk/c3t1d0s6   14    g1   qfs1     on
/dev/dsk/c4t1d0s6   15    g1   qfs1     on
```

2. Use the `mkdir`(1) command to create the `/qfs` mount point for the `/qfs1` file system.

For example:

```
# mkdir /qfs
```

3. Use the `sammkfs`(1M) command to initialize the file system.

   The following example sets the DAU size to 128 kilobytes:

```
# sammkfs -a 128 qfs1
```

4. Use an editor to modify the `/etc/vfstab` file.

   The following example uses the default setting of `stripe=0`, which essentially specifies a round-robin allocation from striped group `g0` to striped group `g1`:

```
qfs1    -    /qfs    samfs    -    yes   stripe=0
```

   This `/etc/vfstab` file sets the stripe width using the `stripe=` option. In this example, there are two striped groups, `g0` and `g1`. With the `stripe=0` specification, devices 12 and 13 are striped, and files are round-robin around the two striped groups. You are really treating a striped group as a bound entity. That is, you cannot change the configuration of the striped group, after it is created, without issuing another `sammkfs`(1M) command.

5. Use the `mount`(1M) command to mount the file system.

   For example:

```
# mount /qfs
```

File System Operations

4

This chapter presents topics related to file system operations. This chapter contains the following sections:

Certain other types of operations and upgrades also need to be performed within StorageTek QFS and StorageTek ASM environments. The following publications describe these other types of operations:

- The *StorageTek ASM Installation and Configuration Guide* describes installing, upgrading, and configuring StorageTek QFS and StorageTek ASM software. It also describes how to create dump files of StorageTek QFS and StorageTek ASM file systems.
- The *StorageTek ASM Storage and Archive Manager Guide* describes how to add slots in an automated library, how to upgrade or replace an automated library, and how to upgrade DLT tape drives.
- The *ASM, ASM-QFS and ASM/ QFS-Standalone Disaster Recovery Guide* describes how to restore StorageTek QFS and StorageTek ASM file systems.

Initializing a File System

You can use the `sammkfs`(1M) command to initialize or reinitialize a StorageTek QFS or StorageTek ASM file system. The following example shows the `sammkfs`(1M) command in its simplest form, with the file system name as its only argument:

```
# sammkfs samqfs1
```

The preceding command builds a version 2 superblock for a standalone StorageTek QFS or StorageTek ASM-QFS file system. The StorageTek QFS and StorageTek ASM file systems support two different superblocks. The StorageTek QFS and StorageTek ASM 4.2 releases support existing file systems with version 1 superblocks but do not allow you to create a version 1 superblock.

For more information about the `sammkfs`(1M) command, its options, and the implications of the version 1 and version 2 superblocks, see "Initializing a File System" on page 43, or see the `sammkfs`(1M) man page. For information about using the `sammkfs`(1M) command to initialize a shared StorageTek QFS file system, see the *StorageTek ASM Installation and Configuration Guide*.

Propagating Configuration File Changes to the System

This section contains procedures that describe how to propagate configuration file changes out to the system. The procedures describe propagating changes for the following files:

- `mcf` file
- `defaults.conf` file
- `archiver.cmd` file (StorageTek ASM-QFS file systems only)
- `stager.cmd` file (StorageTek ASM-QFS file systems only)
- shared hosts file (StorageTek QFS shared and StorageTek ASM-QFS shared file systems only)

You must perform these procedures under the following circumstances:

- If you update any of these files in order to add, delete, or correct information.
- If you create or update an `archiver.cmd`, `defaults.conf`, or `stager.cmd` file after your StorageTek QFS or StorageTek ASM system is already operational.

The following sections describe the procedures:

- "To Change mcf(4) or defaults.conf(4) Information in a StorageTek QFS Environment" on page 55

- “To Change mcf(4) or defaults.conf(4) File System Information in a StorageTek ASM or StorageTek ASM-QFS Environment” on page 56
- “To Change mcf(4) or defaults.conf(4) Removable Media Drive Information” on page 57
- “To Change archiver.cmd(4) or stager.cmd(4) Information” on page 58
- “To Change the Shared Hosts File Information on a Mounted File System” on page 58
- “To Change the Shared Hosts File Information on an Unmounted File System” on page 60

To Change mcf(4) or defaults.conf(4) Information in a StorageTek QFS Environment

To change the `mcf` or the `defaults.conf` information for a shared file system that is configured for high availability in a Sun Cluster environment, perform this procedure on all participating nodes in the Sun Cluster.

1. Use `vi`(1) or another editor to edit the file and change the file system information.
2. Use the `sam-fsd`(1M) command to check the `mcf` file for errors. (Optional)

   Perform this step if you are changing an `mcf` file. For example:

```
# sam-fsd
```

   If the output from this command shows errors, correct them prior to proceeding to the next step.

3. Use the `samd`(1M) `config` command to propagate the `mcf` or `defaults.conf` file changes.

   For example:

```
# samd config
```

   For more information about these files, see the `defaults.conf`(4) or `mcf`(4) man pages.

To Change mcf(4) or defaults.conf(4) File System Information in a StorageTek ASM or StorageTek ASM-QFS Environment

1. Use `vi`(1) or another editor to edit the file and change the file system information.
2. Use the `sam-fsd`(1M) command to check the `mcf` file for errors. (Optional)

   Perform this step if you are changing an `mcf` file. The format of this command is as follows:

```
# sam-fsd
```

   If the output from this command shows errors, correct them prior to proceeding to the next step.

3. Issue a `samcmd`(1M) `aridle` command to idle the archiver for each file system defined in the `mcf` file. (Optional)

   You must perform this step if you are removing or changing information related to one or more file systems. Use this command in the following format:

```
samcmd aridle fs.fsname
```

   For *fsname*, specify the name of a file system defined in the `mcf` file. Issue this command for every file system in the `mcf` file that is affected by the change.

4. Issue a `samcmd`(1M) `idle` command to idle the archiver for each equipment ordinal assigned to a drive in the `mcf` file. (Optional)

   You must perform this step if you are removing or changing information related to one or more drives. Use this command in the following format:

```
samcmd idle eq
```

   For *eq*, specify the Equipment Ordinal of a drive defined in the `mcf` file. Repeat this command as necessary for all drives in your `mcf` file affected by the change.

5. Issue the `umount`(1M) command to unmount the file system(s) affected by the changes.

   For more information about unmounting the file system, see "Unmounting a File System" on page 64.

6. Use the `samd`(1M) `config` command to propagate the changes.

For example:

```
# samd config
```

7. Use the `mount`(1M) command to remount the file system(s) you unmounted.

For more information about these files, see the `defaults.conf`(4) or `mcf`(4) man pages.

To Change mcf(4) or defaults.conf(4) Removable Media Drive Information

1. Edit the file and change the removable media drive information.
2. Use the `sam-fsd`(1M) command to check the `mcf` file for errors. (Optional)

   Perform this step if you are changing an `mcf` file. Use this command in the following format:

```
# sam-fsd
```

   If the output from this command shows errors, correct them prior to proceeding to the next step.

3. Issue a `samcmd`(1M) `aridle` command to idle the archiver for each file system defined in the `mcf` file. (Optional)

   Perform this step if you are removing or changing information related to one or more file systems. Use this command in the following format:

```
samcmd aridle fs.fsname
```

   For *fsname*, specify the name of a file system defined in the `mcf` file. Issue this command for every file system in the `mcf` file that is affected by the change.

4. Issue a `samcmd`(1M) `idle` command for each Equipment Ordinal assigned to a drive in the `mcf` file. (Optional)

   Perform this step if you are removing or changing information related to one or more drives. Use this command in the following format:

```
samcmd idle eq
```

   For *eq*, specify the Equipment Ordinal of a drive defined in the `mcf` file. Repeat this command as necessary for all drives in your `mcf` file affected by the change.

5. Use the `samd`(1M) `stop` command to stop all removable media activity.

   For example:

```
# samd stop
```

6. Use the `samd`(1M) `config` command to propagate the changes and restart the system.

   For example:

```
# samd config
```

7. Use the `samd`(1M) `start` command to start all removable media activity.

   For example:

```
# samd start
```

For more information about these files, see the `defaults.conf`(4) or `mcf`(4) man pages.

To Change archiver.cmd(4) or stager.cmd(4) Information

1. Use `vi`(1) or another editor to edit the `archiver.cmd`(4) or `stager.cmd`(4) file.

2. Use the `archiver`(1M) `-lv` command to validate the changes you made in the `archiver.cmd`(4) file. (Optional)

   Perform this step only if you are changing an existing `archiver.cmd`(4) file.

3. Save and close the file.

4. Use the `samd`(1M) `config` command to propagate the file changes and restart the system.

   For example:

```
# samd config
```

To Change the Shared Hosts File Information on a Mounted File System

Use this procedure to add new host entries or change columns 2 through 5 of the existing shared hosts file entries.

1. If you do not know the host that is acting as the metadata server, issue the `samsharefs`(1M) *file-system-name* command to view the metadata server name.

   Issue this command from any host that has the file system configured.

   For example:

```
# samsharefs sharefs1
```

2. On the metadata server, save the shared hosts file to a temporary working file.

   For example:

```
# samsharefs sharefs1 > /tmp/file
```

3. Save a copy of the shared hosts file. (Optional)

   For example:

```
# cp /tmp/file /var/opt/SUNWsamfs/hosts.date
```

4. Use `vi`(1) or another editor to edit the `/tmp/file` file.

   For mounted file systems, you can add new host entries to the end of the file and you can make changes to columns 2 through 5 of the existing entries.

Note: You cannot change the host names, reorder the entries, or insert entries in the shared hosts file on a mounted file system. To make these changes, unmount the file system on all clients, unmount the metadata server, and then follow the instructions in "To Change the Shared Hosts File Information on an Unmounted File System" on page 60.

5. Save and close the shared hosts file.
6. Copy the new shared hosts file to the `SUNWsamfs` directory.

   For example:

```
# cp /tmp/file /var/opt/SUNWsamfs/hosts.sharefs1
```

7. Apply the new shared hosts file to the file system.

   For example:

```
# samsharefs -u sharefs1
```

To Change the Shared Hosts File Information on an Unmounted File System

Use this procedure to change host names, reorder entries, and insert entries in the shared hosts file.

1. If you do not know the host that is acting as the metadata server, issue the `samsharefs`(1M) `-R` *file-system-name* command to view the metadata server name.

   Issue this command from any host that has the file system configured.

   For example:

```
# samsharefs -R sharefs1
```

2. Unmount the file system on each participating client, and then on the metadata server.
3. On the metadata server, save the shared hosts file to a temporary working file.

   For example:

```
# samsharefs -R sharefs1 > /tmp/file
```

4. Save a copy of the shared hosts file. (Optional)

   For example:

```
# cp /tmp/file /var/opt/SUNWsamfs/hosts.date
```

5. Use `vi`(1) or another editor to edit the `/tmp/file` file.

   For unmounted file systems, you can change host names, reorder entries, insert entries, add new host entries, and edit columns 2 through 5 of existing entries.
6. Save and close the shared hosts file.
7. Copy the new shared hosts file to the `SUNWsamfs` directory.

   For example:

```
# cp /tmp/file /var/opt/SUNWsamfs/hosts.sharefs1
```

8. Apply the new shared hosts file to the file system.

For example:

```
# samsharefs -uR sharefs1
```

9. Mount the file system on the metadata server, and then on the clients.

Mounting a File System

You can mount a StorageTek QFS or StorageTek ASM file system by using the Solaris OS `mount`(1M) command. This section describes the various ways that mount options can be specified.

Mount parameters are used to manipulate file system characteristics. There are several ways to specify mount parameters, and some specification methods override others. You can specify mount options in the following ways:

1. With the `mount`(1M) command using command line options. Highest priority. Options specified on the Solaris OS `mount`(1M) command line override other options specified in the `/etc/vfstab` file, directives specified in the `samfs.cmd` file, and system default settings.
2. As `/etc/vfstab` file settings. Second priority.
3. In the `samfs.cmd` file using directives. Third priority.
4. System defaults. Fourth (lowest) priority. The default system settings are the configurable settings already defined for your Solaris OS. You can override the system settings with specifications in the `samfs.cmd` file, in the `/etc/vfstab` file, and on the `mount`(1M) command.

You can also specify mount options by using the the `samu`(1M) operator utility or the `samcmd`(1M) command. Mount options enabled or disabled by using any of these utilities persist until the file system is unmounted.

The following sections describe the ways to specify mount options in more detail, explain when to use these files and commands, and show the order in which they take precedence. In addition to the following sections, the *StorageTek ASM Installation and Configuration Guide* includes information about mounting a file system.

The mount(1M) Command

The Solaris OS `mount`(1M) command mounts the file system and enables you to specify settings that override the settings specified in the `/etc/vfstab` file and in the `/etc/opt/SUNWsamfs/samfs.cmd` file. For example, you can specify the stripe width, readahead, writebehind, high and low water marks for disk cache utilization, and so on.

One way to use the `mount`(1M) command in conjunction with the `samfs.cmd` file is to use the `samfs.cmd` file as your main location for mount options and to use options on the `mount`(1M) command when experimenting with or tuning your system. The `mount`(1M) command options override both the `/etc/vfstab` entries and the directives in the `samfs.cmd` file.

Example. The following command mounts file system `qfs1` at `/work` with `setuid` execution disallowed and `qwrite` enabled. The `qfs1` file system name is the Equipment Identifier. This also appears in the `mcf` file's Equipment Identifier field for this file system. To specify more than one mount option, separate each with a comma.

```
# mount -o nosuid,qwrite qfs1 /work
```

If you are mounting a StorageTek QFS or StorageTek ASM-QFS shared file system, you must mount the file system on the metadata server first, and then mount the file system on each participating client host. Include the shared option with the `mount` command, and keep in mind that the command is identical on the metadata server and on the participating hosts.

For more information about the `mount`(1M) command, see the `mount_samfs`(1M) man page.

The /etc/vfstab File

The `/etc/vfstab` Solaris OS system file must contain a line for each StorageTek QFS or StorageTek ASM file system that is defined in the `mcf` file. This is required to mount the file system. For each file system, you must provide information for the seven fields shown in Table 18..

Table 18. Fields in the `/etc/vfstab` File

Field Number	Content
1	The file system family set name.
2	The file system to `samfsck`(1M).
3	The mount point.
4	The file system type. This is always `samfs`, even for StorageTek QFS file systems.
5	The `samfsck`(1M) pass.
6	Mount at boot options.
7	Mount parameters.

The fields in the `/etc/vfstab` file must be separated by either space or tab characters. The mount parameters in the seventh field, however, must each be separated by a comma character (,) without any intervening spaces.

Example. The following is an example of an /etc/vfstab file.

```
qfs1    -    /qfs    samfs    -    yes    stripe=0
```

The mount parameters field can contain any of the mount parameters listed as arguments to the -o option on the mount_samfs(1M) man page. These parameters are nearly identical to those that you can specify as directive lines in the samfs.cmd file or as arguments to the -o option on the mount(1M) command. As with the samfs.cmd file, you can include specifications for various I/O settings, readahead, writebehind, the stripe width, various Application Storage Manager (ASM) settings, Qwrite, and other features.

For more information about possible mount parameters, see the mount_samfs(1M) man page. For more information about modifying the /etc/vfstab file, see the vfstab(4) man page.

The samfs.cmd File

The /etc/opt/SUNWsamfs/samfs.cmd file enables you to specify mount parameters for all your StorageTek QFS and StorageTek ASM file systems. This file can be useful when you have multiple file systems configured and you want to specify the same mount parameters for them.

The list of possible mount parameters is very comprehensive. The possible mount parameters you can specify pertain to I/O settings, readahead, writebehind, the stripe width, various Application Storage Manager (ASM) settings, Qwrite, and other features.

Using this file enables you to define all your mount parameters in one place in an easily readable format. Directives specified toward the beginning of this file are global directives, and they apply to all StorageTek QFS and StorageTek ASM file systems. The second part of this file enables you to indicate the specific parameters that you want to apply to each individual file system. The ability to specify the common parameters once, and only in one place, differentiates this file from the /etc/vfstab file, in which you must specify all mount parameters for each file system in the seventh field.

The mount parameters that can be specified in the samfs.cmd file are nearly identical to those that you can specify in the /etc/vfstab file or as arguments to the -o option on the mount(1M) command. For more information about the mount parameters that can be specified in this file, see the samfs.cmd(4) man page.

In the samfs.cmd file, the directives are written one per line. The file can contain comments, which must begin with a pound character (#). Characters that appear to the right of the pound character are treated as comments.

Directives that appear before any fs = line apply globally to all file systems. A line that starts with fs = must precede directives that are specific to a particular file system. Directives specific to a particular file system override global directives.

Figure 26. is a sample `samfs.cmd` file that sets the low and high water marks for disk cache utilization and specifies individualized parameters for two file systems.

Figure 26. Example `samfs.cmd` File

```
low = 50
high = 75
fs = samfs1
   high = 65
   writebehind = 512
   readahead = 1024
fs = samfs5
   partial = 64
```

The directives in the `samfs.cmd` file serve as defaults and override any default system settings, but arguments to the `mount`(1M) command override any directives in this file. Entries in the `/etc/vfstab` file also override directives specified in the `samfs.cmd` file.

For information about which directives can be entered in the `samfs.cmd` file, see the `samfs.cmd`(4) man page. For information about the `mount`(1M) command, see the `mount_samfs`(1M) man page.

■ Unmounting a File System

You can use the Solaris OS `umount`(1M) command to unmount StorageTek QFS and StorageTek ASM file systems.

On StorageTek ASM and StorageTek ASM-QFS file systems, you must issue commands to stop the archiver prior to unmounting the file system. The following procedure shows you how to idle the archiver and unmount the file system. You do not need to idle the archiver if you are using a StorageTek QFS file system.

To Unmount Standalone StorageTek ASM and StorageTek ASM-QFS File Systems

1. Issue a `samcmd`(1M) `aridle fs.`*fsname* command for the file system. (Optional)

   Perform this step if you are unmounting a StorageTek ASM or StorageTek ASM-QFS file system. For example:

```
# samcmd aridle fs.samqfs2
```

This step in the procedure cleanly halts the archiving for file system `samqfs2`. Specifically, it allows archiving operations to halt at a logical place before stopping the daemons.

2. Issue a `samd`(1M) `stop` command. (Optional)

   For example:

```
# samd stop
```

3. Unmount the file system.

```
# umount /samqfs
```

   Several conditions can be present in a file system at unmounting time, so you might need to issue the `umount`(1M) command a second time. If the file system still does not unmount, use `unshare`(1M), `fuser`(1M), or other commands in conjunction with the `umount`(1M) command. Unmounting procedures are also described in the *StorageTek ASM Installation and Configuration Guide*.

To Unmount StorageTek QFS and StorageTek ASM-QFS Shared File Systems

The `umount`(1M) command unmounts a shared file system from a Solaris system. For more information about the `umount`(1M) command, see the `umount`(1M) man page.

1. Use the `umount`(1M) command to unmount the file system on every participating client host.

```
# umount /samqfs
```

2. On the metadata server, halt the archiving for the file system at a logical place by issuing the `samcmd`(1M) `aridle fs.`*fsname* command. (Optional)

   Perform this step in a StorageTek ASM-QFS file system environment.

   For example:

```
# samcmd aridle fs.samqfs2
```

   This example halts the archiving for file system `samqfs2`.

3. Unmount the file system on the metadata server.

```
# umount /samqfs
```

Unmount the metadata server only after unmounting all client hosts.

Several conditions can be present in a file system at unmounting time, so you might need to issue the `umount`(1M) command a second time. If the file system still does not unmount, use `unshare`(1M), `fuser`(1M), or other commands in conjunction with the `umount`(1M) command. Unmounting procedures are also described in the *StorageTek ASM Installation and Configuration Guide*.

Checking File System Integrity and Repairing File Systems

StorageTek QFS and StorageTek ASM file systems write validation records in the following records that are critical to file system operations: directories, indirect blocks, and inodes. If the file system detects corruption while searching a directory, it issues an EDOM error, and the directory is not processed. If an indirect block is not valid, it issues an ENOCSI error, and the file is not processed. Table 19. summarizes these error indicators.

Table 19. Error Indicators

Error	Solaris OS Meaning	StorageTek QFS and StorageTek ASM Meaning
EDOM	Argument is out of domain.	Values in validation records are out of range.
ENOCSI	No CSI structure is available.	Links between structures are invalid.

In addition, inodes are validated and cross checked with directories.

You should monitor the following files for error conditions:

- The log file specified in `/etc/syslog.conf` for the errors shown in Table 19..
- The `/var/adm/messages` file for device errors.

If a discrepancy is noted, you should unmount the file system and check it using the `samfsck`(1M) command.

Note: The `samfsck`(1M) command can be issued on a mounted file system, but the results cannot be trusted. Because of this, you are encouraged to run the command on an unmounted file system only.

To Check a File System

1. Use the `samfsck`(1M) command to perform a file systems check.

   Use this command in the following format:

```
samfsck -V family_set_name
```

For *family_set_name*, specify the name of the file system as specified in the `mcf` file.

You can send output from `samfsck`(1M) to both your screen and to a file by using it in conjunction with the `tee`(1) command, as follows.

- C shell:

```
# samfsck -V family_set_name |& tee file
```

- Bourne shell:

```
# samfsck -V family_set_name 2>&1 | tee file
```

Nonfatal errors returned by `samfsck`(1M) are preceded by `NOTICE`. Nonfatal errors are lost blocks and orphans. The file system is still consistent if `NOTICE` errors are returned. You can repair these nonfatal errors during a convenient, scheduled maintenance outage.

Fatal errors are preceded by `ALERT`. These errors include duplicate blocks, invalid directories, and invalid indirect blocks. The file system is not consistent if these errors occur. Notify StorageTek if the `ALERT` errors cannot be explained by a hardware malfunction.

If the `samfsck`(1M) command detects file system corruption and returns `ALERT` messages, you should determine the reason for the corruption. If hardware is faulty, repair it prior to repairing the file system.

For more information about the `samfsck`(1M) and `tee`(1) commands, see the `samfsck`(1M) and `tee`(1) man pages.

To Repair a File System

1. Use the `umount`(1M) command to unmount the file system.

   Run the `samfsck`(1M) command when the file system is not mounted. For information about unmounting a file system, see "Unmounting a File System" on page 64.

2. Use the `samfsck`(1M) command to repair a file system. If you are repairing a shared file system, issue the command from the metadata server.

You can issue the `samfsck`(1M) command in the following format to repair a file system:

```
# samfsck -F -V fsname
```

For *fsname*, specify the name of the file system as specified in the `mcf` file.

Preserving Information for an Upgrade

If you are about to add or change disks, controllers, or other equipment in your environment, it can be difficult to correct or regenerate all the file system descriptions in the `mcf` file. The `samfsconfig`(1M) command can help you by generating information about your file system and file system components after making these changes.

The `samfsconfig`(1M) command examines devices and determines if any of them have StorageTek QFS or StorageTek ASM superblocks on them. It uses information from the discovered superblocks and aggregates the devices into a format similar to an `mcf` file. You can save this format and edit it to recreate a damaged, missing, or incorrect `mcf` file.

This command returns information about each device that you specify and writes this information to `stdout`. The command can retrieve the family set number of the base device (the file system itself), the file system type (`ma` or `ms`), and whether the file system is a StorageTek QFS shared file system.

Irregularities are flagged with one of the following:

- A pound sign (`#`). This indicates incomplete family set information.
- A greater-than sign (`>`). This indicates that more than one device name refers to a particular file system element.

If necessary, this command's output can be used to help regenerate the file system portions of your `mcf` file if your system is reconfigured or experiences a disaster. The following examples show output from the `samfsconfig`(1M) command.

Example 1

In this example, the system administrator has put a list of device names into a file. These device names are not accounted for in the environment. The system administrator wants to examine only these devices for

StorageTek QFS and StorageTek ASM family sets. The results show some old fragments of family sets and several complete instances.

Figure 27. Example 1 - Output From `samfsconfig`**(1M) Command**

```
mn# samfsconfig -v `cat /tmp/dev_files`
Device '/dev/dsk/c0t0d0s0' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c0t0d0s1' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c0t0d0s3' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c0t0d0s4' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c0t0d0s5' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c0t0d0s6' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c0t0d0s7' doesn't have a SAM-FS superblock (SBLK).
Couldn't open '/dev/dsk/c0t1d0s0';  errno=5.
Couldn't open '/dev/dsk/c0t1d0s1';  errno=5.
Device '/dev/dsk/c0t1d0s3' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c0t1d0s4' has a SAM-FS superblock.
Device '/dev/dsk/c0t1d0s5' has a SAM-FS superblock.
Device '/dev/dsk/c0t1d0s6' has a SAM-FS superblock.
Couldn't open '/dev/dsk/c0t1d0s7';  errno=5.
Couldn't open '/dev/dsk/c0t6d0s0';  errno=16.
Couldn't open '/dev/dsk/c0t6d0s1';  errno=16.
Couldn't open '/dev/dsk/c0t6d0s3';  errno=16.
Couldn't open '/dev/dsk/c0t6d0s4';  errno=16.
Couldn't open '/dev/dsk/c0t6d0s5';  errno=16.
Couldn't open '/dev/dsk/c0t6d0s6';  errno=16.
Couldn't open '/dev/dsk/c0t6d0s7';  errno=16.
Couldn't open '/dev/dsk/c1t0d0s3';  errno=5.
Couldn't open '/dev/dsk/c1t0d0s4';  errno=5.
Couldn't open '/dev/dsk/c1t0d0s5';  errno=5.
Device '/dev/dsk/c1t0d0s6' doesn't have a SAM-FS superblock (SBLK).
Couldn't open '/dev/dsk/c1t0d0s7';  errno=5.
Couldn't open '/dev/dsk/c1t1d0s0';  errno=2.
Couldn't open '/dev/dsk/c1t2d0s3';  errno=5.
Couldn't open '/dev/dsk/c1t2d0s4';  errno=5.
Couldn't open '/dev/dsk/c1t2d0s5';  errno=5.
Device '/dev/dsk/c1t2d0s6' doesn't have a SAM-FS superblock (SBLK).
Couldn't open '/dev/dsk/c1t2d0s7';  errno=5.
Could not read from device '/dev/dsk/c1t3d0s0'; errno=5.
Couldn't open '/dev/dsk/c1t4d0s3';  errno=5.
Couldn't open '/dev/dsk/c1t4d0s4';  errno=5.
Couldn't open '/dev/dsk/c1t4d0s5';  errno=5.
Device '/dev/dsk/c1t4d0s6' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c1t4d0s7' doesn't have a SAM-FS superblock (SBLK).
Couldn't open '/dev/dsk/c1t5d0s3';  errno=5.
Couldn't open '/dev/dsk/c1t5d0s4';  errno=5.
Couldn't open '/dev/dsk/c1t5d0s5';  errno=5.
Device '/dev/dsk/c1t5d0s6' doesn't have a SAM-FS superblock (SBLK).
Couldn't open '/dev/dsk/c1t5d0s7';  errno=5.
Device '/dev/dsk/c3t0d0s0' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c3t0d0s1' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c3t0d0s3' has a SAM-FS superblock.
```

Figure 27. Example 1 - Output From `samfsconfig`(1M) Command (Continued)

```
Device '/dev/dsk/c3t0d0s4' has a SAM-FS superblock.
Couldn't open '/dev/dsk/c3t0d0s7';  errno=5.
Device '/dev/dsk/c3t1d0s0' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c3t1d0s1' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c3t1d0s3' has a SAM-FS superblock.
Device '/dev/dsk/c3t1d0s4' has a SAM-FS superblock.
Couldn't open '/dev/dsk/c3t1d0s7';  errno=5.
Device '/dev/dsk/c4t0d0s0' has a SAM-FS superblock.
Could not read from device '/dev/dsk/c4t0d0s1'; errno=5.
Could not read from device '/dev/dsk/c4t0d0s3'; errno=5.
Could not read from device '/dev/dsk/c4t0d0s4'; errno=5.
Could not read from device '/dev/dsk/c4t0d0s5'; errno=5.
Device '/dev/dsk/c4t0d0s6' has a SAM-FS superblock.
Device '/dev/dsk/c4t0d0s7' has a SAM-FS superblock.
Device '/dev/dsk/c4t1d0s0' has a SAM-FS superblock.
Could not read from device '/dev/dsk/c4t1d0s1'; errno=5.
Could not read from device '/dev/dsk/c4t1d0s3'; errno=5.
Could not read from device '/dev/dsk/c4t1d0s4'; errno=5.
Could not read from device '/dev/dsk/c4t1d0s5'; errno=5.
Device '/dev/dsk/c4t1d0s6' has a SAM-FS superblock.
Device '/dev/dsk/c4t1d0s7' has a SAM-FS superblock.
Device '/dev/dsk/c4t2d0s0' has a SAM-FS superblock.
Could not read from device '/dev/dsk/c4t2d0s1'; errno=5.
Could not read from device '/dev/dsk/c4t2d0s3'; errno=5.
Could not read from device '/dev/dsk/c4t2d0s4'; errno=5.
Could not read from device '/dev/dsk/c4t2d0s5'; errno=5.
Device '/dev/dsk/c4t2d0s6' has a SAM-FS superblock.
Device '/dev/dsk/c4t2d0s7' has a SAM-FS superblock.
Device '/dev/dsk/c4t3d0s0' has a SAM-FS superblock.
Could not read from device '/dev/dsk/c4t3d0s1'; errno=5.
Could not read from device '/dev/dsk/c4t3d0s3'; errno=5.
Could not read from device '/dev/dsk/c4t3d0s4'; errno=5.
Could not read from device '/dev/dsk/c4t3d0s5'; errno=5.
Device '/dev/dsk/c4t3d0s6' has a SAM-FS superblock.
Device '/dev/dsk/c4t3d0s7' has a SAM-FS superblock.
19 SAM-FS devices found.
#
# Family Set 'samfs2' Created Mon Jun 25 10:37:52 2001
#
# Missing slices
# Ordinal 1
# /dev/dsk/c0t1d0s6    12    md   samfs2  -
#
# Family Set 'samfs1' Created Wed Jul 11 08:47:38 2001
#
# Missing slices
# Ordinal 1
# /dev/dsk/c0t1d0s4    12    md   samfs1  -
# Ordinal 2
# /dev/dsk/c0t1d0s5    13    md   samfs1  -
#
```

Figure 27. Example 1 - Output From `samfsconfig`(1M) Command (Continued)

```
# Family Set 'samfs2' Created Sat Nov  3 17:22:44 2001
#
samfs2 ma 30 samfs2 - shared
/dev/dsk/c4t0d0s6    31    mm   samfs2  -
/dev/dsk/c4t1d0s6    32    mr   samfs2  -
/dev/dsk/c4t2d0s6    33    mr   samfs2  -
#
# Family Set 'qfs1' Created Wed Nov  7 15:16:19 2001
#
qfs1 ma 10 qfs1 -
/dev/dsk/c3t0d0s3    11    mm   qfs1  -
/dev/dsk/c3t0d0s4    12    g0   qfs1  -
/dev/dsk/c3t1d0s3    13    g0   qfs1  -
/dev/dsk/c3t1d0s4    14    g0   qfs1  -
#
# Family Set 'sharefsx' Created Wed Nov  7 16:55:19 2001
#
sharefsx ma 200 sharefsx - shared
/dev/dsk/c4t0d0s0    210    mm   sharefsx  -
/dev/dsk/c4t1d0s0    220    mr   sharefsx  -
/dev/dsk/c4t2d0s0    230    mr   sharefsx  -
/dev/dsk/c4t3d0s0    240    mr   sharefsx  -
#
# Family Set 'samfs5' Created Tue Nov 27 16:32:28 2001
#
samfs5 ma 80 samfs5 -
/dev/dsk/c4t3d0s6    82    mm   samfs5  -
/dev/dsk/c4t3d0s7    83    g0   samfs5  -
/dev/dsk/c4t0d0s7    84    g0   samfs5  -
/dev/dsk/c4t1d0s7    85    g1   samfs5  -
/dev/dsk/c4t2d0s7    86    g1   samfs5  -
```

Example 2

In this example, the devices flagged with a greater-than sign (>) are duplicated. The `s0` slice starts at the start of disk, as does the whole disk (`s2`) slice. This is the style of output obtained in a Solaris 9 OS.

Figure 28. shows the `samfsconfig`(1M) command and output.

Figure 28. Example 2 - Output from `samfsconfig` Command

```
# samfsconfig /dev/dsk/c3t*
#
# Family Set 'shsam1' Created Wed Oct 17 14:57:29 2001
#
shsam1 160 ma shsam1 shared
> /dev/dsk/c3t50020F23000055A8d0s2    161    mm   shsam1  -
> /dev/dsk/c3t50020F23000055A8d0s0    161    mm   shsam1  -
/dev/dsk/c3t50020F23000055A8d0s1    162    mr   shsam1  -
```

Figure 28. Example 2 - Output from `samfsconfig` Command

```
> /dev/dsk/c3t50020F23000078F1d0s0    163    mr    shsam1   -
> /dev/dsk/c3t50020F23000078F1d0s2    163    mr    shsam1   -
/dev/dsk/c3t50020F23000078F1d0s1    164    mr    shsam1   -
```

Example 3

In this example, the whole disk slice (slice 2) is left off of the command line. This is the style of output obtained in a Solaris 9 OS.

Figure 29. shows the `samfsconfig`(1M) command and output.

Figure 29. Example 3 - Output from `samfsconfig`(1M) Command

```
# samfsconfig /dev/dsk/c3t*s[013-7]
#
# Family Set 'shsam1' Created Wed Oct 17 14:57:29 2001
#
shsam1 160 ma shsam1 shared
/dev/dsk/c3t50020F23000055A8d0s0    161    mm    shsam1   -
/dev/dsk/c3t50020F23000055A8d0s1    162    mr    shsam1   -
/dev/dsk/c3t50020F23000078F1d0s0    163    mr    shsam1   -
/dev/dsk/c3t50020F23000078F1d0s1    164    mr    shsam1   -
```

For more information about this command, see the `samfsconfig`(1M) man page.

■ Preparing for a Hardware Device Upgrade

Whether upgrading a server, adding a new tape drive, adding an automated library, or installing a different drive into an existing automated library, it is best to plan in advance. This section prepares you for hardware upgrades to devices within your environment.

StorageTek recommends the following actions prior to the upgrade:

- Determine if the hardware addition or change requires a new license from StorageTek.

  Examples of changes that do not require a license upgrade include adding memory and increasing disk cache. Examples of changes that require a license upgrade include adding more slots in an automated library and changing the model of your server.

- Read the hardware manufacturer's installation instructions carefully. Also read the documentation on adding hardware in your Solaris OS system administrator documentation.

- Check the Equipment Ordinals between your old and new `mcf` files. For information about the `mcf` file, see the `mcf`(4) man page.

- Decide whether or not the backup copies you have on hand are sufficient. For information about backing up your data and metadata, see the procedures described in the *StorageTek ASM Installation and Configuration Guide*.
  - In a StorageTek QFS environment, the `qfsdump`(1M) command dumps all data and metadata. For more information about this process, see the `qfsdump`(1M) man page.
  - In StorageTek ASM and StorageTek ASM-QFS environments, the `samfsdump`(1M) command dumps all metadata. You must ensure that all files that need to be archived have an archive copy. Use the `archive_audit`(1) command on each StorageTek ASM or StorageTek ASM-QFS file system to see which files do not have an archive copy. In the following example, `/sam` is the mount point.

```
# archive_audit /sam
```

- Ensure that the system is quiet with no users logged in.
- In StorageTek ASM and StorageTek ASM-QFS environments, ensure that the archiver is in `wait` mode. The archiver must be in `wait` mode, and not running, during an upgrade.

  You can idle the archiver in one of the following ways:
  - By inserting a `wait` directive into the `/etc/opt/SUNWsamfs/archiver.cmd` file. For more information about the `wait` directive and the `archiver.cmd` file, see the `archiver.cmd`(4) man page.
  - By using the `samu`(1M) operator utility.
  - By issuing the following command:

```
# samcmd aridle
```

  For more information, see the `samcmd`(1M) man page.

Adding Disk Cache to a File System

At some point, you might want to add disk partitions or disk drives in order to increase the disk cache for a file system. You accomplish this by updating the `mcf` file and using the `samgrowfs`(1M) command. You do not need to reinitialize or restore the file system.

In StorageTek ASM and StorageTek ASM-QFS environments, note that when adding disks or partitions, the system might update the Equipment Ordinal of the historian. The system automatically generates the Equipment Ordinal of the historian unless you specifically call it out. For more information, see the `historian`(7) man page.

To Add Disk Cache to a File System

1. Use the `umount`(1M) command to unmount the file system you want to expand. If the file system is shared, unmount the file system on all client hosts and then on the metadata server. You can then perform the remaining steps in this procedure on the metadata server.

   For information about unmounting a file system, see "Unmounting a File System" on page 64.

2. If you want to rename the file system during this procedure, use the `samfsck`(1M) command with its `-R` and `-F` options to rename the file system. (Optional)

   For more information about this command, see the `samfsck`(1M) man page.

3. Edit the `/etc/opt/SUNWsamfs/mcf` file.

   You can configure up to 252 disk partitions in a file system.

   To increase the size of a StorageTek QFS file system, at least one new metadata partition must be added. Metadata partitions require an Equipment Type of `mm`. Zero or more data partitions can be added.

   If you want to add new partitions for metadata or for data, add them to the `mcf` file after the existing disk partitions. Save the changes, and quit the editor.

   Do not change the Equipment Identifier name in the `/etc/opt/SUNWsamfs/mcf` file. If the name in the `mcf` file does not match the name in the superblock, the file systems can no longer be mounted. Instead, the following message is logged in `/var/adm/messages`:

```
WARNING SAM-FS superblock equipment identifier <id>s on eq
<eq> does not match <id> in mcf
```

4. Type the `sam-fsd`(1M) command to check for errors in the `mcf` file.

   For example:

```
# sam-fsd
```

   If the `sam-fsd`(1M) command output indicates that there are errors in the `mcf` file, fix them before proceeding to the next step in this procedure.

5. Type the `samd`(1M) `config` command to propagate the `mcf` file changes to the system.

For example:

```
# samd config
```

For more information about the `samd`(1M) command, see the `samd`(1M) man page.

6. Type the `samgrowfs`(1M) command on the file system that is being expanded.

   For example, type the following command to expand file system `samfs1`:

```
# samgrowfs samfs1
```

   If you renamed your file system, run the `samgrowfs`(1M) command on the new name. For more information about this command, see the `samgrowfs`(1M) man page.

7. Mount the file system.

   If you renamed your file system, enter the new name in the `mcf` file and then issue the `sam-fsd(1M)` and the `samd(1M) config` commands as described in steps 4 and 5.

8. For StorageTek QFS shared file systems only, edit the `mcf` file on each participating client host to match the metadata server's `mcf` file.

Replacing Disks in a File System

At some point, you might want to perform the following tasks:

- Change disks or partitions
- Add disks or partitions
- Remove disks or partitions

To accomplish these tasks, you need to back up and recreate the file system by following the steps in this procedure.

To Back Up and Recreate a File System

1. Back up all site-customized system files and configuration files.

   Depending on your software, these files can include `mcf`, `archiver.cmd`, `defaults.conf`, `samfs.cmd`, `inquiry.conf`, and so on. Back up these files for all file systems in your StorageTek QFS and StorageTek ASM environments. Also make sure that you have backup copies of files in the `/etc/opt/SUNWsamfs` directory, files in the `/var/opt/SUNWsamfs` directory, library catalogs, the historian, shared hosts files, and any parameter files for network-attached automated libraries.

In StorageTek ASM and StorageTek ASM-QFS environments, if you do not know the names and locations of your catalog files, examine the `mcf` file with `vi`(1) or another viewing command and find the first `rb` entry in the `mcf` file. That entry contains the name of the library catalog file. If no catalog file location is specified, then the system is using the default location (`/var/opt/SUNWsamfs/catalog`).

2. Ensure that each file system to be modified is backed up.

   The file systems should be backed up regularly according to your site's policies. This is described as the last step in the installation procedure. If you are comfortable with the backup files that already exist for your file systems, there is no need to back them up again now. If, however, you need to back up your file systems to preserve information created since the last dump file was created, do so now. For information about how to create a dump file, see the *StorageTek ASM Installation and Configuration Guide*.

   Note that if you are using the StorageTek ASM or StorageTek ASM-QFS file system, the `samfsdump`(1M) command issues warnings when creating the dump file if it encounters unarchived files in the file system. If warnings are issued, these files need to be archived before unmounting the file systems.

3. Unmount the file system.

   For information about unmounting a file system, see "Unmounting a File System" on page 64.

4. If you want to rename the file system during this procedure, use the `samfsck`(1M) command with its `-R` and `-F` options to rename the file system. (Optional)

   For more information about this command, see the `samfsck`(1M) man page.

5. Edit the `/etc/opt/SUNWsamfs/mcf` file.

   You can configure up to 252 disk partitions in a file system. Edit the `mcf` file to add or delete disks or partitions. New partitions must be added after existing disk partitions. Save the changes, and quit the editor.

   To increase the size of a StorageTek QFS file system, at least one new metadata partition must be added. Metadata partitions require an Equipment Type of `mm`. Zero or more data partitions can be added.

   Do not change the Equipment Identifier name in the `/etc/opt/SUNWsamfs/mcf` file. If the name in the `mcf` file does not match the name in the superblock, the file systems can no longer be mounted. Instead, the following message is logged in /var/adm/messages:

```
WARNING SAM-FS superblock equipment identifier <id>s on eq
<eq> does not match <id> in mcf
```

6. Type the sam-fsd(1M) command to check for errors in the mcf file.

   For example:

```
# sam-fsd
```

   If the sam-fsd(1M) command output indicates that there are errors in the mcf file, fix them before proceeding to the next step in this procedure.

   For more information, see the sam-fsd(1M) man page.

7. Type the samd(1M) config command to propagate the mcf file changes.

   For example:

```
# samd config
```

   For more information, see the samd(1M) man page.

8. Type the sammkfs(1M) command to make a new file system.

   For example, the following command creates samfs10:

```
# sammkfs samfs10
```

9. Type the mount(1M) command to mount the file system.

   For information about mounting a StorageTek QFS or StorageTek ASM file system, see the mount_samfs(1M) man page.

10. Type the cd(1) command to change to the mount point of the file system.

11. Use the samfsrestore(1M) or qfsrestore(1M) command to restore each file.

    Restore from the the dump file you had or from the dump file created in Step 2.

    For information about using these commands, see the samfsdump(1M) or qfsdump(1M) man pages, or see the *ASM, ASM-QFS and ASM/QFS-Standalone Disaster Recovery Guide*.

12. Use the restore.sh(1M) script to stage back all files that had been online.

Use this command in the following format:

```
# restore.sh log_file mount_point
```

For *log_file*, specify the name of the log file that was created by the `sammkfs`(1M) or the `samfsrestore`(1M) commands.

For *mount_point*, specify the mount point of the file system being restored.

For information about the `restore.sh`(1M) script, see the `restore.sh`(1M) man page.

Upgrading a Host System

When it comes time to upgrade the host system being used for the file system, take the following into account:

- It is wise to move to the new host while the existing host is still in operation. This allows time to install, configure, and test the new hardware platform with your applications.
- Moving to a new host system is equivalent to installing the StorageTek QFS or StorageTek ASM software for the first time. In StorageTek ASM and StorageTek ASM-QFS environments, you need to reinstall the software and update the configuration files (specifically the `mcf` file, the `/kernel/drv/st.conf` file, and the `/etc/opt/SUNWsamfs/inquiry.conf` file). In addition, you need to copy your existing `archiver.cmd` and `defaults.conf` files to the new system, configure system logging, and so on.

  You can use the installation instructions in the *StorageTek ASM Installation and Configuration Guide* when re-installing the software.
- You might need to update your license key. License keys are tied to the CPU host ID. Replacing the system requires a new license.
- Before powering down the old host system, decide whether or not the backup copies you have on hand are sufficient. You might need to create new dump files at this time. A dump file is used to recreate the file system on the new server. For more information about creating a dump file, see the `qfsdump`(1M) or `samfsdump`(1M) man pages or see the *StorageTek ASM Installation and Configuration Guide*.

Upgrading the Solaris OS

The following sections describe how to upgrade your Solaris OS:

- "To Upgrade Your Solaris OS in a StorageTek ASM or StorageTek ASM-QFS Environment" on page 79

- “To Upgrade Your Solaris OS in a StorageTek QFS Environment” on page 82

To Upgrade Your Solaris OS in a StorageTek ASM or StorageTek ASM-QFS Environment

Many of the steps involved in upgrading your Solaris OS level are identical to the steps involved in upgrading your StorageTek ASM or StorageTek ASM-QFS environment. Some of the steps in this procedure reference procedures in the *StorageTek ASM Installation and Configuration Guide*.

1. Obtain the software upgrade.

   The StorageTek ASM and StorageTek ASM-QFS software supports various levels of the Solaris OS. You should not reinstall your old StorageTek ASM or StorageTek ASM-QFS software onto your newly upgraded Solaris OS.

   In addition, depending on the revision level currently installed and the level to which you are upgrading, you might need a new software license.

   Contact StorageTek or your authorized service provider to obtain new copies of the software and to determine whether or not your site needs a new license.

2. Back up all site-customized system files and configuration files.

   These files include `mcf`, `archiver.cmd`, `defaults.conf`, `samfs.cmd`, `inquiry.conf`, and so on. Back up these files for all file systems in your StorageTek ASM and StorageTek ASM-QFS environments.

   Also make sure that you have backup copies of files in the `/etc/opt/SUNWsamfs` directory, files in the `/var/opt/SUNWsamfs` directory, library catalogs, the historian, and any parameter files for network-attached automated libraries.

   If you do not know the names and locations of your catalog files, examine the `mcf` file with `vi`(1) or another viewing command and find the first `rb` entry in the `mcf` file. That entry contains the name of the library catalog file. If no catalog file location is specified, then the system is using the default location (`/var/opt/SUNWsamfs/catalog`).

3. Ensure that each file system affected is backed up.

   The file systems should be backed up regularly according to your site’s policies. This is one of the the last steps in the installation procedure. If you are comfortable with the backup files that already exist for your file systems, there is no need to back them up again now. If, however, you need to back up your file systems to preserve information created since the last dump file was created, do so now.

Note that if you are using the StorageTek ASM or StorageTek ASM-QFS file system, the `samfsdump`(1M) command issues warnings when creating the dump file if it encounters unarchived files in the file system. If warnings are issued, these files need to be archived before unmounting the file systems.

4. Unmount the file systems.

   For information about unmounting a file system, see "Unmounting a File System" on page 64.

5. Issue a `samd`(1M) `stop` command to stop all archiving and ASM activity. (Optional)

   Perform this step if you are unmounting a StorageTek ASM or StorageTek ASM-QFS file system. For example:

```
# samd stop
```

6. Remove existing StorageTek ASM or StorageTek ASM-QFS software.

   Use the `pkgrm`(1M) command to remove the existing software. You must remove all existing StorageTek ASM and StorageTek ASM-QFS packages before installing either the new packages or the new operating system level.

   For example, the following command removes the `SUNWsamtp`, `SUNWsamfsu`, and the `SUNWsamfsr` packages in a StorageTek ASM or StorageTek ASM-QFS environment. The `SUNWsamfsr` package must be removed last. Note that the `SUNWsamtp` package is an optional tools package, and it might not be installed on your system. An example `pkgrm`(1M) command is as follows:

```
# pkgrm SUNWsamtp SUNWsamfsu SUNWsamfsr
```

   The information in this step assumes that you are removing software packages at the 4.2 release level or later. The software package names changed as of the 4.2 releases. If you have earlier releases of the software packages on your system, see the *StorageTek ASM Installation and Configuration Guide* for information about removing them.

7. Upgrade the Solaris OS.

   Install the new Solaris OS revision using the Sun Solaris upgrade procedures for the OS level you are installing.

8. Add the `SUNWsamfsr` and `SUNWsamfsu` packages that you obtained in Step 1.

   The StorageTek ASM and StorageTek ASM-QFS software packages use the Solaris OS packaging utilities for adding and deleting software. You must be logged in as superuser to make changes to software packages. The `pkgadd`(1M) command prompts you to confirm various actions necessary to upgrade the StorageTek ASM and StorageTek ASM-QFS package.

On the installation CD-ROM, the StorageTek ASM and StorageTek ASM-QFS package resides in the `/cdrom/cdrom0` directory.

Run the `pkgadd`(1M) command, as follows, to upgrade the packages, answering `yes` to each question:

```
# pkgadd -d . SUNWsamfsr SUNWsamfsu
```

During the installation, the system detects the presence of conflicting files and prompts you to indicate whether or not you want to continue with the installation. You can go to another window and copy the files you wish to save to an alternate location.

9. Update the license keys. (Optional)

   Depending on the StorageTek ASM and StorageTek ASM-QFS software revision you had, and the revision to which you are upgrading, you might need to obtain new license keys for your software. Contact your StorageTek or your authorized service provider for help on determining if you need a new license.

   If you are upgrading from a release prior to 4.2, you need to place a new license key in the following file:

```
/etc/opt/SUNWsamfs/LICENSE.4.2
```

   For more information, see the licensing information in the *StorageTek ASM Installation and Configuration Guide*.

10. Mount the file system(s). (Optional)

   You must perform this step if you have not modified the `/etc/vfstab` file to have `yes` in the Mount at Boot field.

   Use the `mount`(1M) command to mount the file systems and continue operation with the upgraded StorageTek ASM or StorageTek ASM-QFS software.

   In the following example, `samfs1` is the name of the file system to be mounted.

```
# mount samfs1
```

To Upgrade Your Solaris OS in a StorageTek QFS Environment

Many of the steps involved in upgrading your Solaris OS level are identical to the steps involved in upgrading your StorageTek QFS environment. Some of the steps in this procedure reference procedures in the *StorageTek ASM Installation and Configuration Guide*.

1. Obtain the software upgrade.

   StorageTek QFS software supports various levels of the Solaris OS. You should not reinstall your old StorageTek QFS software onto your newly upgraded Solaris OS.

   In addition, depending on the revision level currently installed and the level to which you are upgrading, you may need a new StorageTek QFS license.

   Contact your StorageTek or your authorized service provider to obtain new copies of the software and to determine whether or not your site needs a new license.

2. Back up all site-customized system files and configuration files.

   These files include `mcf`, `defaults.conf`, `samfs.cmd`, the shared hosts files, and so on. Back up these files for all file systems in your StorageTek QFS environment. Also make sure that you have backup copies of files in the `/etc/opt/SUNWsamfs` directory.

3. Ensure that each file system affected is backed up.

   The file systems should be backed up regularly according to your site's policies. This is described as the last step in the installation procedure. If you are comfortable with the backup files that already exist for your file systems, there is no need to back them up again now. If, however, you need to back up your file systems to preserve information created since the last dump file was created, do so now. For information about how to create a dump file, see the *StorageTek ASM Installation and Configuration Guide*.

4. Unmount the file systems.

   For information about unmounting a file system, see "Unmounting a File System" on page 64.

5. Remove existing StorageTek QFS software.

   Use the `pkgrm`(1M) command to remove the existing software. You must remove the existing StorageTek QFS package before installing either the new package or the new operating system level.

For example, the following command removes the StorageTek QFS software:

```
# pkgrm SUNWqfsu SUNWqfsr
```

The information in this step assumes that you are removing a software package at the 4.2 release level or later. The software package names changed as of the 4.2 releases. If you have earlier releases fo the software packages on your system, see the *StorageTek ASM Installation and Configuration Guide* for information about removing them.

6. Upgrade the Solaris OS.

   Install the new Solaris OS revision using the Sun Solaris upgrade procedures for the Solaris OS level you are installing.

7. Add the packages that you obtained in Step 1.

   The StorageTek QFS software package uses the Solaris OS packaging utilities for adding and deleting software. You must be logged in as superuser to make changes to software packages. The `pkgadd`(1M) command prompts you to confirm various actions necessary to upgrade the StorageTek QFS package.

   On the installation CD-ROM, the StorageTek QFS package resides in the `/cdrom/cdrom0` directory.

   Run the `pkgadd`(1M) command to upgrade the package, answering `yes` to each question:

```
# pkgadd -d . SUNWqfsr SUNWqfsu
```

   During the installation, the system detects the presence of conflicting files and prompts you to indicate whether or not you want to continue with the installation. You can go to another window and copy any files you want to save to an alternate location.

8. Update the license keys. (Optional)

   Depending on the StorageTek QFS software revision you had, and the revision to which you are upgrading, you might need to obtain new license keys for your StorageTek QFS software. Contact your StorageTek or your authorized service provider for help on determining if you need a new license.

   If you are upgrading from a StorageTek QFS release prior to 4.2, you need to place a new license key in the following file:

```
/etc/opt/SUNWsamfs/LICENSE.4.2
```

For more information, see the licensing information in the *StorageTek ASM Installation and Configuration Guide*.

9. Update the `mcf` file. (Optional)

   If device names have changed, it might be necessary to update the `mcf` file to match the new device names. Veryify the new device names, and then follow the procedure in “Propagating Configuration File Changes to the System” on page 54.

10. Mount the file system(s). (Optional)

    Perform this step if you have not modified the `/etc/vfstab` file to have `yes`.

    Use the procedure described in “Mounting a File System” on page 61. Continue operation with the upgraded StorageTek QFS software.

StorageTek QFS Shared File System

5

A StorageTek QFS shared file system is a distributed file system that can be mounted on multiple Solaris operating system (OS) host systems. In a StorageTek QFS shared file system environment, one Solaris OS host acts as the metadata server for the file system, and additional hosts can be configured as clients. You can configure more than one host as a potential metadata server, but only one host can be the metadata server at any one time. There is no limit to the number of StorageTek QFS shared file system mount points.

The advantage of the StorageTek QFS shared file system is that file data passes directly from the Fibre Channel disks to the hosts. Data travels via local path I/O (also known as *direct access I/O*). This is in contrast to the Network File System (NFS), which transfers data over the network.

This chapter describes how to configure and maintain the StorageTek QFS shared file system. This chapter contains the following sections:

Overview

In a StorageTek QFS shared file system environment, no archiving or staging occurs, so no network connection to archive media is necessary. If you are operating a StorageTek QFS shared file system environment, ignore the information about archive media in this chapter.

In a StorageTek ASM-QFS shared file system environment, each host that can become the metadata server needs to be connected to the same archive media repository, which can be one of the following:

- A library with removable media devices (tape or magneto-optical drives).
- Disk space in one or more file systems.

  You must specify the archive media in the `mcf` file or in the `diskvols.conf` file on each host that can become a metadata server.

  In a StorageTek ASM-QFS shared file system environment, the active metadata server is the only host upon which the staging (`sam-stagerd`) and archiving (`sam-archiverd`) daemons are active. The metadata server is designated as the server from which all file requests are staged.

  You cannot configure a StorageTek ASM-QFS shared file system in a Sun Cluster environment.

This chapter describes how to maintain a StorageTek QFS shared file system. It assumes that you have installed the StorageTek QFS or StorageTek ASM-QFS software on the host systems according to the instructions in the *StorageTek ASM Installation and Configuration Guide*.

Note: The StorageTek QFS shared file system cannot be configured in a StorageTek ASM (an `ms` file system) environment.

Configuring the StorageTek QFS Shared File System

To perform initial installation and configuration for a StorageTek QFS shared file system, follow the instructions in the *StorageTek ASM Installation and Configuration Guide*. Many examples in this chapter use host names and configuration information that were introduced in that guide.

Converting an Unshared File System to a Shared File System

To convert an unshared StorageTek QFS file system to a StorageTek QFS shared file system, perform the following procedures in the order shown.

To Convert an Unshared File System to Shared on the Server

1. As superuser, log in to the system to be used as the primary metadata server.

   You must have `root` permission to complete the steps in this procedure.

2. Back up all site-customized system files and configuration files.

   Depending on your software, these files can include `mcf`, `archiver.cmd`, `defaults.conf`, `samfs.cmd`, `inquiry.conf`, and so on. Back up these files for all file systems. Also make sure that you have backup copies of files in the `/etc/opt/SUNWsamfs` directory, files in the `/var/opt/SUNWsamfs` directory, library catalogs, the historian, and any parameter files for network-attached automated libraries.

   In StorageTek ASM-QFS environments, if you do not know the names and locations of your catalog files, look for the automated libraries in the Additional Parameters field of the `mcf` file. If the Additional Parameters field is blank, however, the system uses the default path of `/var/opt/SUNWsamfs/catalog/`*catalog_name*. For more information about catalog file locations, see the `mcf`(4) man page.

3. Ensure that each file system to be modified is backed up. (Optional)

   If you want to move files from an existing StorageTek QFS or StorageTek ASM-QFS file system into a new StorageTek QFS shared file system, make sure that your file systems are backed up. The file systems should be backed up regularly according to your site's policies. This is described as the last step in the installation procedure. If you are comfortable with the backup files that already exist for your file systems, there is no need to back them up again now.

4. Use the `umount`(1M) command to unmount the file system.

5. Use the `sammkfs`(1M) `-F -S` *fsname* command to convert the file system to a StorageTek QFS shared file system.

   For *fsname*, specify the Family Set Name of the file system that you are converting to a new StorageTek QFS shared file system. For example:

```
# sammkfs -S -F sharefs1
```

6. Edit the `/etc/opt/SUNWsamfs/mcf` file to add the `shared` keyword in the file system's Additional Parameters field.

For example:

Figure 30. `mcf` **File for Shared File System,** `sharefs1`

```
# Equipment                        Eq   Eq    Family    Dev   Add
# Identifier                      Ord Type Set      State Params
# ----------                      --- ---- ------   ----- ------
sharefs1                          10  ma   sharefs1 on    shared
/dev/dsk/c2t50020F23000065EEd0s6 11  mm   sharefs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12  mr   sharefs1 on
/dev/dsk/c7t50020F2300006099d0s6 13  mr   sharefs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14  mr   sharefs1 on
```

7. Edit the `/etc/vfstab` file to add the `shared` keyword in the file system's Mount Parameters field.

   For example:

Figure 31. `/etc/vfstab` **File Example**

```
# File /etc/vfstab
# FS name   FS to fsck    Mnt pt    FS type   fsck pass   Mt@boot   Mt
params
sharefs1    -            /sharefs1 samfs     -           no        shared
```

8. Create the `/etc/opt/SUNWsamfs/hosts.`*fsname* hosts configuration file.

   For example:

Figure 32. StorageTek QFS Shared File System Hosts File Example

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host   Host IP                          Server   Not  Server
# Name   Addresses                        Priority Used Host
# ----   -------------------------------- -------- ---- ---
--
host1     172.16.0.129,titan.xyzco.com      1        -
server
host2     172.16.0.130,tethys.xyzco.com     2        -
```

9. Run the `samsharefs`(1M) `-u -R` *fsname* command to initialize the file system and the host configuration.

   For example:

```
# samsharefs -u -R sharefs1
```

10. Run the `samd`(1M) `config` command.

This informs the `sam-fsd` daemon of the configuration changes. For example:

```
# samd config
```

11. Issue the `mount`(1M) command to mount the file system.

To Convert an Unshared File System to Shared on Each Client

1. Use the `mkdir`(1) command to create the mount point for the file system.

   For example:

```
# mkdir /sharefs1
```

2. Create an `/etc/opt/SUNWsamfs/hosts.`*fsname*`.local` local hosts configuration file. (Optional)

Figure 33. File `hosts.sharefs1.local`

```
# This is file /etc/opt/SUNWsamfs/hosts.sharefs1.local
# Host Name    Host Interfaces
# ---------    ---------------
titan          172.16.0.129
tethys         172.16.0.130
```

3. Ensure that each file system to be modified is backed up. (Optional)

   If you want to move files from an existing StorageTek QFS or StorageTek ASM-QFS file system into a new StorageTek QFS shared file system, make sure that your file systems are backed up. The file systems should be backed up regularly according to your site's policies. This is described as the last step in the installation procedure. If you are comfortable with the backup files that already exist for your file systems, there is no need to back them up again now.

4. Use the `umount`(1M) command to unmount the file system.

5. Edit the `/etc/vfstab` file to add the `shared` keyword in the file system's Mount Parameters field.

   For example:

Figure 34. `/etc/vfstab` **File Example**

```
# File /etc/vfstab
# FS name   FS to fsck    Mnt pt    FS type   fsck pass   Mt@boot   Mt params
sharefs1    -             /sharefs1 samfs     -            no         shared
```

6. Create the `/etc/opt/SUNWsamfs/hosts.`*fsname* hosts configuration file.

   For example:

Figure 35. StorageTek QFS Shared File System Hosts File Example

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host   Host IP                          Server   Not  Server
# Name   Addresses                        Priority Used Host
# ----   -------------------------------- -------- ---- -----
host1     172.16.0.129,titan.xyzco.com     1         -    server
host2     172.16.0.130,tethys.xyzco.com    2         -
```

Converting a Shared File System to an Unshared File System

To convert a StorageTek QFS shared file system to an unshared StorageTek QFS file system, perform the following procedures in the order shown.

To Convert a Shared File System to Unshared on Each Client

1. Use the `umount`(1M) command to unmount the file system.
2. Delete the file system's entry from the `/etc/opt/SUNWsamfs/mcf` file.
3. Delete the file system's entry from the `/etc/vfstab` file.
4. Run the `samd`(1M) `config` command.

   This informs the `sam-fsd` daemon of the configuration changes. For example:

```
# samd config
```

5. Delete the mount point for the file system.

To Convert a Shared File System to Unshared on the Server

1. As superuser, log in to the metadata server system.

   You must have `root` permission to complete the steps in this procedure.

2. Back up all site-customized system files and configuration files.

   Depending on your software, these files can include `mcf`, `archiver.cmd`, `defaults.conf`, `samfs.cmd`, `inquiry.conf`, and so on. Back up these files for all file systems. Also make sure that you have backup copies of files in the `/etc/opt/SUNWsamfs` directory, files in the `/var/opt/SUNWsamfs` directory, library catalogs, the historian, and any parameter files for network-attached automated libraries.

In StorageTek ASM-QFS environments, if you do not know the names and locations of your catalog files, examine the `mcf` file with `vi`(1) or another viewing command and find the entries for the automated libraries. The path to each library's catalog files is in the Additional Parameters field. If the Additional Parameters field is blank, however, the system uses the default path of `/var/opt/SUNWsamfs/catalog/`*catalog_name*. For more information about catalog file locations, see the `mcf`(4) man page.

3. Ensure that each file system to be modified is backed up. (Optional)

   If you want to move files from an existing StorageTek QFS shared or StorageTek ASM-QFS shared file system into a new StorageTek QFS file system, make sure that your file systems are backed up. The file systems should be backed up regularly according to your site's policies. This is described as the last step in the installation procedure. If you are comfortable with the backup files that already exist for your file systems, there is no need to back them up again now.

4. Use the `umount`(1M) command to unmount the file system.

5. Run the `sammkfs`(1M) `-F -U` *fsname* to convert the StorageTek QFS shared file system to an unshared file system.

   For *fsname*, specify the Family Set Name of the StorageTek QFS shared file system that you are converting to a new unshared file system. For example:

```
# sammkfs -F -U samfs1
```

6. Edit the `/etc/opt/SUNWsamfs/mcf` file to remove the `shared` keyword from the file system's Addtional Parameters field.

   For example:

Figure 36. `mcf` **File for File System,** `samfs1`

```
# Equipment                          Eq  Eq    Family   Dev   Add
# Identifier                        Ord Type Set      State Params
# ----------                        --- ---- ------   ----- ------
samfs1                               10  ma   samfs1   on
/dev/dsk/c2t50020F23000065EEd0s6 11  mm   samfs1   on
/dev/dsk/c7t50020F2300005D22d0s6 12  mr   samfs1   on
/dev/dsk/c7t50020F2300006099d0s6 13  mr   samfs1   on
/dev/dsk/c7t50020F230000651Cd0s6 14  mr   samfs1   on
```

7. Edit the `/etc/vfstab` file to remove the `shared` keyword from the file system's Mount Parameters field.

For example:

Figure 37. `/etc/vfstab` **File Example**

```
# File /etc/vfstab
# FS name    FS to fsck     Mnt pt     FS type    fsck pass    Mt@boot    Mt
params
samfs1     -              /samfs1      samfs      -             no
```

8. Delete the `/etc/opt/SUNWsamfs/hosts.`*fsname* configuration file.

9. Run the `samd`(1M) `config` command.

   This informs the `sam-fsd` daemon of the configuration changes. For example:

```
# samd config
```

10. Use the `mount`(1M) command to mount the file system.

Mounting and Unmounting StorageTek QFS Shared File Systems

When mounting or unmounting a StorageTek QFS shared file system, the order in which the Solaris OS is mounted or unmounted is important.

For failover purposes, the mount options should be the same on the metadata server and all potential metadata servers. For example, you can create a `samfs.cmd`(4) file containing mount options and copy it to all the hosts.

For more information about mounting StorageTek QFS shared file systems, see “Mount Options in a StorageTek QFS Shared File System” on page 108 or see the `mount_samfs`(1M) man page. For more information about mounting and unmounting file systems, see “File System Operations” on page 53.

To Mount a StorageTek QFS Shared File System

The `mount`(1M) command mounts a StorageTek QFS shared file system in a Solaris OS. For more information about the `mount`(1M) command, see the `mount`(1M) man page.

1. Become superuser on the metadata server and on all the client hosts.

2. Use the `mount`(1M) command to mount the metadata server.

   Mount the file system on the metadata server prior to mounting it on any client hosts.

3. Use the `mount`(1M) command to mount the client hosts.

   You can mount the file system on the client hosts in any order.

To Unmount a StorageTek QFS Shared File System

Follow the instructions in "To Unmount StorageTek QFS and StorageTek ASM-QFS Shared File Systems" on page 65.

■ Adding and Removing a Client Host

The following sections describe adding and removing client host systems:

- "To Add a Client Host" on page 93
- "To Remove a Client Host" on page 102

To Add a Client Host

You can add a client host to a StorageTek QFS shared file system after you have configured and mounted the file system on all participants. If you are adding a client host that is a node in a Sun Cluster environment, you must add the node to the cluster's existing resource group. For more information, see the *Sun Cluster System Administration Guide for Solaris OS*.

Follow these steps for each client host.

1. Become superuser on the metadata server.
2. Use the `samsharefs`(1M) command to retrieve the current StorageTek QFS shared file system information and write it to an editable file.

- If the StorageTek QFS shared file system is mounted, issue the `samsharefs`(1M) command on the current metadata server. For example:

```
# samsharefs sharefs1 > /etc/opt/SUNWsamfs/hosts.sharefs1
```

- If the StorageTek QFS shared file system is unmounted, issue the `samsharefs`(1M) command with its `-R` option from the metadata server or from any of the potential metadata servers. For example:

```
# samsharefs -R sharefs1 > /etc/opt/SUNWsamfs/
hosts.sharefs1
```

  You can issue the `samsharefs`(1M) command only on the active metadata server or on client hosts configured as potential metadata servers. For more information, see the `samsharefs`(1M) man page.

Note: You can change the hosts information on any potential metadata server when the file system is unmounted. StorageTek recommends that you always retrieve the hosts information to ensure that the hosts information is current.

3. Use `vi`(1) or another editor to open the StorageTek QFS shared file system information file.

   Figure 38. shows this step.

Figure 38. `hosts.sharefs1` **Prior to Editing**

```
# vi /etc/opt/SUNWsamfs/hosts.sharefs1
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host    Host IP                              Server    Not   Server
# Name    Addresses                             Priority Used Host
# ----    --------------------------------- -------- ---- ---
--
titan     172.16.0.129,titan.xyzco.com          1        -
server
tethys    172.16.0.130,tethys.xyzco.com         2        -
mimas     mimas.xyzco.com                       -        -
dione     dione.xyzco.com                       -        -
```

4. Use the editor to add a line for the new client host.

   Figure 39. shows the file after adding the line for `helene` as the last line.

Figure 39. `hosts.sharefs1` **After Editing**

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host    Host IP                              Server    Not   Server
# Name    Addresses                             Priority Used Host
# ----    --------------------------------- -------- ---- ---
--
titan     172.16.0.129,titan.xyzco.com          1        -
server
tethys    172.16.0.130,tethys.xyzco.com         2        -
mimas     mimas.xyzco.com                       -        -
dione     dione.xyzco.com                       -        -
helene    helene.xyzco.com                      -        -
```

5. Use the `samsharefs`(1M) command to update the current information in the binary file.

   The options to use on this command, and the system from which it is issued, differ depending on whether or not the StorageTek QFS shared file system is mounted, as follows:

- If the StorageTek QFS shared file system is mounted, issue the `samsharefs`(1M) `-u` command from the current metadata server. For example:

```
# samsharefs -u sharefs1
```

- If the StorageTek QFS shared file system is unmounted, issue the `samsharefs`(1M) `-R -u` command from the active metadata server or from any of the potential metadata servers. For example:

```
# samsharefs -R -u sharefs1
```

The client host `helene` is now recognized.

6. As superuser, log in to one of the client hosts.
7. Use the `format`(1M) command to verify the presence of client host disks.
8. Update the `mcf` file on the client host.

Before a host system can access or mount a shared file system, it must have that file system defined in its `mcf` file.

Use `vi`(1) or another editor to edit the `mcf` file on one of the client host systems. The `mcf` file must be updated on all client hosts to be included in the StorageTek QFS shared file system. The file system and disk declaration information must have the same data for the Family Set Name, Equipment Ordinal, and Equipment Type as the configuration on the metadata server. The `mcf` files on the client hosts must also include the shared keyword. The device names, however, can change as controller assignments can change from host to host.

The `samfsconfig`(1M) command generates configuration information that can help you to identify the devices included in the StorageTek QFS shared file system. Enter a separate `samfsconfig`(1M) command on each client host. Note that the controller number might not be the same controller number as on the metadata server because the controller numbers are assigned by each client host.

Example 1. Figure 40. shows how the `samfsconfig`(1M) command is used to retrieve device information for family set `sharefs1` on client `tethys`. Note that `tethys` is a potential metadata server, so it is connected to the same metadata disks as `titan`.

Figure 40. `samfsconfig`**(1M) Command Example on** `tethys`

```
tethys# samfsconfig /dev/dsk/*
#
# Family Set 'sharefs1' Created Wed Jun 27 19:33:50 2003
#
sharefs1                         10 ma sharefs1 on shared
```

Figure 40. `samfsconfig`**(1M) Command Example on** `tethys` **(Continued)**

```
/dev/dsk/c2t50020F23000065EEd0s6 11 mm sharefs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12 mr sharefs1 on
/dev/dsk/c7t50020F2300006099d0s6 13 mr sharefs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14 mr sharefs1 on
```

Edit the `mcf` file on client host `tethys` by copying the last five lines of output from the `samfsconfig`(1M) command into the `mcf` file on client host `tethys`. Verify the following:

- Each Device State field is set to `on`.
- The `shared` keyword appears in the Additional Parameters field for the file system name.

Figure 41. shows the resulting `mcf` file.

Figure 41. `mcf` **File for** `sharefs1` **Client Host** `tethys`

```
# Equipment                         Eq   Eq    Family    Dev   Add
# Identifier                       Ord Type Set       State Params
# ----------                       --- ---- ------    ----- ------
sharefs1                           10  ma   sharefs1 on    shared
/dev/dsk/c2t50020F23000065EEd0s6 11   mm   sharefs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12   mr   sharefs1 on
/dev/dsk/c7t50020F2300006099d0s6 13   mr   sharefs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14   mr   sharefs1 on
```

In Figure 41., note that the Equipment Ordinal numbers match those of the example `mcf` file for metadata server `titan`. These Equipment Ordinal numbers must not already be in use on client host `tethys` or any other client host.

Example 2. Figure 42. shows how the `samfsconfig`(1M) command is used to retrieve device information for family set `sharefs1` on client host `mimas`. Note that `mimas` can never become a metadata server, and it is not connected to the metadata disks.

Figure 42. `samfsconfig`**(1M) Command Example on** `mimas`

```
mimas# samfsconfig /dev/dsk/*
#
# Family Set 'sharefs1' Created Wed Jun 27 19:33:50 2001
#
# Missing slices
# Ordinal 0
```

Figure 42. `samfsconfig`**(1M) Command Example on** `mimas` **(Continued)**

```
# /dev/dsk/c1t50020F2300005D22d0s6   12    mr    sharefs1   on
# /dev/dsk/c1t50020F2300006099d0s6   13    mr    sharefs1   on
# /dev/dsk/c1t50020F230000651Cd0s6   14    mr    sharefs1   on
```

In the output from the `samfsconfig`(1M) command on `mimas`, note that `Ordinal` 0, which is the metadata disk, is not present. Because devices are missing, the `samfsconfig`(1M) command comments out the elements of the file system and omits the file system Family Set declaration line. Make the following types of edits to the `mcf` file:

- Create a file system Family Set declaration line, beginning with `sharefs1`, in the `mcf` file for client host `mimas`. Enter the `shared` keyword in the Additional Parameters field of the file system Family Set declaration line.
- Create one or more `nodev` lines for each missing Equipment Ordinal. For these lines, the keyword `nodev` must appear in the Equipment Identifier field for each inaccessible device. In this example, you create a device entry in the `mcf` file named `nodev` to represent the missing metadata disk.
- Ensure that each Device State field is set to `on`.
- Uncomment the device lines.

Figure 43. shows the resulting `mcf` file for `mimas`.

Figure 43. `mcf` **File for Client Host** `mimas`

```
# The mcf File For mimas
# Equipment                          Eq   Eq    Family    Device Addl
# Identifier                          Ord Type Set         State
Params
------------                          --- ---- ---         -----   --
----
sharefs1                              10  ma   sharefs1 on
shared
nodev                                 11  mm   sharefs1 on
/dev/dsk/c1t50020F2300005D22d0s6 12  mr   sharefs1 on
/dev/dsk/c1t50020F2300006099d0s6 13  mr   sharefs1 on
/dev/dsk/c1t50020F230000651Cd0s6 14  mr   sharefs1 on
```

Note: If you update a metadata server's `mcf` file after the StorageTek ASM-QFS shared file system is mounted, be sure to update the `mcf` files on all hosts that can access that shared file system.

9. Issue the `samd`(1M) `config` command on the metadata server host.

   This informs the `sam-fsd` daemon of the configuration changes. For example:

```
# samd config
```

10. Create the local hosts configuration file on the client host. (Optional)

You might want to perform this step if your StorageTek QFS shared host systems have multiple host interfaces. The local hosts configuration file defines the host interfaces that the metadata server and the client hosts can use when accessing the file sysem. You use this file to specify how file system traffic should flow over public and private networks in your environment.

The local hosts configuration file must reside in the following location:

```
/etc/opt/SUNWsamfs/hosts.fsname.local
```

For *fsname*, specify the Family Set Name of the StorageTek QFS shared file system.

Comments are permitted in the local hosts configuration file. Comment lines must begin with a pound character (#). Characters to the right of the pound character are ignored.

Table 20. shows the fields in the local hosts configuration file.

Table 20. Local Hosts Configuration File Fields

Field Number	Content
1	The Host Name field. This field must contain the alphanumeric name of a metadata server or potential metadata server that is part of the StorageTek QFS shared file system.
2	The Host Interfaces field. This field must contain a comma-separated list of host interface addresses. This field can be created by using the output received from the `ifconfig`(1M) `-a` command. The individual interfaces can be specified in one of the following ways: • Dotted-decimal IP address form • IP version 6 hexadecimal address form • As a symbolic name that the local domain name service (DNS) can resolve to a particular host interface Each host uses this field to determine whether a host will try to connect to the specified host interface. The system evaluates the addresses from left to right, and the connection is made using the first responding address in the list that is also included in the shared hosts file.

In a StorageTek QFS shared file system, each client host obtains the list of metadata server IP addresses from the metadata server host.

The metadata server and the client hosts use both the `/etc/opt/SUNWsamfs/hosts.`*fs_name* file on the metadata server and the `hosts.`*fsname*`.local` file on each client host (if it exists) to determine the host interface to use when accessing the file system. This process is as follows (note that *client*, as in *network client*, is used to refer to both client hosts and the metadata server host in the following process):

1. The client obtains the list of metadata server host IP interfaces from the file system's on-disk host file. To examine this file, issue the `samsharefs`(1M) command from the metadata server or from a potential metadata server.
2. The client searches its files for a `hosts.`*fsname*`.local` file. Depending on the outcome of the search, one of the following courses of action is taken:
   - If a `hosts.`*fsname*`.local` file does not exist, the client attempts to connect, in turn, to each address in the system hosts configuration file until it succeeds in connecting.
   - If the `hosts.`*fsname*`.local` file exists, the client performs the following tasks:
     a. It compares the list of addresses for the metadata server from both the `/etc/opt/SUNWsamfs/hosts.`*fsname* file on the metadata server and the `hosts.`*fsname*`.local` file.
     b. It builds a list of addresses that are present in both places, and then it attempts to connect to each of these addresses, in turn, until it succeeds in connecting to the server. If the order of the addresses differs in these files, the client uses the ordering in the `hosts.`*fsname*`.local` file.

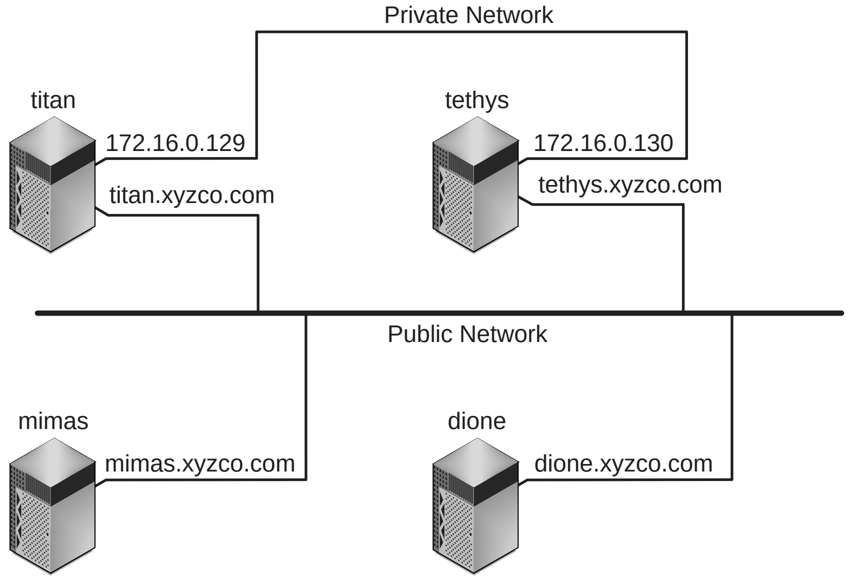
Example. This example expands on the example that was introduced in the *StorageTek ASM Installation and Configuration Guide*. Figure 44. is an example hosts file that shows four hosts.

Figure 44. StorageTek QFS Shared File System Hosts File Example

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host    Host IP                           Server   Not  Server
# Name    Addresses                         Priority Used Host
# ----    -------------------------------- -------- ---- ---
--
titan     172.16.0.129,titan.xyzco.com      1        -
server
tethys    172.16.0.130,tethys.xyzco.com     2        -
mimas     mimas.xyzco.com                   -        -
dione     dione.xyzco.com                   -        -
```

Figure 45. shows the interfaces to these systems.

Figure 45. Network Interfaces



Systems `titan` and `tethys` share a private network connection with interfaces `172.16.0.129` and `172.16.0.130`. To guarantee that `titan` and `tethys` always communicate over their private network connection, the system administrator has created identical copies of `/etc/opt/SUNWsamfs/hosts.sharefs1.local` on each system. Figure 46. shows the information in these files.

Figure 46. File `hosts.sharefs1.local` on Both `titan` and `tethys`

```
# This is file /etc/opt/SUNWsamfs/hosts.sharefs1.local
# Host Name     Host Interfaces
# ---------     ---------------
titan           172.16.0.129
tethys          172.16.0.130
```

Systems `mimas` and `dione` are not on the private network. To guarantee that they connect to `titan` and `tethys` through `titan`'s and `tethys`'s public interfaces, and never attempt to connect to `titan`'s or `tethys`'s unreachable private interfaces, the system administrator has created identical copies of `/etc/opt/SUNWsamfs/hosts.sharefs1.local` on `mimas` and `dione`. Figure 47. shows the information in these files.

Figure 47. File `hosts.sharefs1.local` on Both `mimas` and `dione`

```
# This is file /etc/opt/SUNWsamfs/hosts.sharefs1.local
# Host Name      Host Interfaces
# ----------     --------------
titan           titan.xyzco.com
tethys          tethys.xyzco.com
```

3. **Issue the `samd`(1M) `config` command** on the client host.

This informs the sam-fsd daemon of the configuration changes. For example:

```
# samd config
```

4. Verify that the sam-sharefsd daemon is running for this file system.

   To acomplish this, use the ps(1) and grep(1) commands as shown in Figure 48..

Figure 48. Output from the ps(1) Command

```
# ps -ef | grep sam-sharefsd
root 26167 26158  0 18:35:20 ?         0:00 sam-sharefsd
sharefs1
root 27808 27018  0 10:48:46 pts/21    0:00 grep sam-sharefsd
```

   Figure 48. shows that the sam-sharefsd daemon is active for the sharefs1 file system. If this is the case for your system, you can proceed to the next step in this procedure. If, however, the output returned on your system does not show that the sam-sharefsd daemon is active for your StorageTek QFS shared file system, perform the diagnostic procedures described in "Recovering a Hung mount(1M) Command" on page 120.

5. If your mount point does not exist, make the mount point for the new StorageTek QFS shared file system.

   Use the mkdir(1) command to make the directory for the mount point. For example:

```
# mkdir /sharefs1
```

6. Issue the chmod(1M) command to give the mount point the 755 set of permissions.

   For example:

```
# chmod 755 /sharefs1
```

   The permissions must be the same on all participant hosts. 755 is suggested as the initial permission set because users must have execute permission on the mount point in order to be able to use the file system after it has been mounted. After you mount the file systems, the root directory's permissions override this setting.

7. Modify the /etc/vfstab file.

   You must have an entry in the /etc/vfstab file for the StorageTek QFS shared file system. Specify shared in the Mount Parameters field.

If you want the StorageTek QFS shared file system to automatically mount at boot, make the following changes in the `/etc/vfstab` file:

- Type yes in the `Mt@boot` field.
- Add the `bg` mount option in the `Mt params` field. The `bg` mount option mounts the file system in the background if the metadata server is not responding.

If you do not want to mount this file system automatically at boot time, type `no` in the `Mt@boot` field. In either case, as Figure 49. shows, `shared` is a required entry in the `Mt params` field.

Figure 49. `/etc/vfstab` **File Example**

```
# File /etc/vfstab
# FS name   FS to fsck   Mnt pt    FS type   fsck   Mt@boot   Mt
params
#                                            pass
sharefs1    -            /sharefs1 samfs     -      yes
shared,bg
```

8. Issue the `df`(1M) command on the metadata server to verify that the file system is mounted on the metadata server.

   For example:

```
# df -k
```

9. From the client host, issue the `mount`(1M) command to mount the StorageTek QFS shared file system on the client host.

   For failover purposes, the mount options should be the same on the metadata server and all potential metadata servers. For example, you can create a `samfs.cmd`(4) file containing mount options and copy it to all the hosts.

   For more information about mounting StorageTek QFS shared file systems, see “Mount Options in a StorageTek QFS Shared File System” on page 108, or see the `mount_samfs`(1M) man page.

   For example:

```
# mount /sharefs1
```

To Remove a Client Host

1. Become superuser on the metadata server and on all the client hosts.

   Tip – You can use the `samsharefs`(1M) command to verify that you are, indeed, logged into the metadata server or a client host.

2. Use the `umount`(1M) command to unmount the StorageTek QFS shared file system on the first client host.

   Repeat this step for all client hosts that have the StorageTek QFS shared file system mounted.

   For example:

```
client# umount sharefs1
```

3. Use the `umount`(1M) command to unmount the StorageTek QFS shared file system on the metadata server.

   For example:

```
metaserver# umount sharefs1
```

4. If you have not already done so, log in as superuser to the metadata server for the StorageTek QFS shared file system.

5. Use the `samsharefs`(1M) command to obtain the current configuration information.

   The following example command writes current configuration information to file `/etc/opt/SUNWsamfs/hosts.sharefs1`:

```
# samsharefs -R sharefs1 > /etc/opt/SUNWsamfs/
hosts.sharefs1
```

6. Use `vi`(1) or another editor to open the StorageTek QFS shared file system information file.

   Figure 50. shows the file prior to deleting the client host.

Figure 50. `hosts.sharefs1` **Prior to Deleting a Client Host**

```
# vi /etc/opt/SUNWsamfs/hosts.sharefs1
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host   Host IP                          Server   Not  Server
# Name   Addresses                        Priority Used Host
# ----   -------------------------------- -------- ---- -----
titan    172.16.0.129,titan.xyzco.com     1        -    server
tethys   172.16.0.130,tethys.xyzco.com    2        -
mimas    mimas.xyzco.com                  -        -
dione    dione.xyzco.com                  -        -
helene   helene.xyzco.com                 -        -
```

7. Use the editor to delete the client host or hosts that are no longer to be supported.

Figure 51. shows the file after the line for `helene` has been deleted.

Figure 51. `hosts.sharefs1` **After Deleting a Client Host**

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host    Host IP                          Server   Not  Server
# Name    Addresses                        Priority Used Host
# ----    -------------------------------- -------- ---- -----
titan     172.16.0.129,titan.xyzco.com     1        -    server
tethys    172.16.0.130,tethys.xyzco.com    2        -
mimas     mimas.xyzco.com                  -        -
dione     dione.xyzco.com                  -        -
```

8. Use the `samsharefs`(1M) `-R -u` command to update the current hosts information.

   For example:

```
# samsharefs -R -u sharefs1
```

   The host `helene` has been removed.

9. Use the `samsharefs`(1M) `-R` command to display the current configuration.

   For example:

```
# samsharefs -R sharefs1
```

10. Use the `mount`(1M) command to mount the StorageTek QFS shared file system on the metadata server.

    For information about the `mount`(1M) command, see the `mount_samfs`(1M) man page.

11. Use the `mount`(1M) command to mount the StorageTek QFS shared file system on the client hosts.

    For information about the `mount`(1M) command, see the `mount_samfs`(1M) man page.

Changing the Metadata Server in a StorageTek QFS Environment

The procedures in the following sections describe how to change which host is acting as the metadata server in a StorageTek QFS shared file system without using the automatic Membership Services feature of a software package such as Sun Cluster.

You can change the metadata server system manually under the following circumstances:

- If the metadata server becomes unavailable
- If you want to change the metadata server or the potential metadata servers

For changing the metadata server to succeed, the mount options of the existing metadata server and all potential metadata servers must be the same.

Note: Contact the StorageTek Professional Services Group if you need assistance in changing the metadata server in a StorageTek ASM-QFS environment.

Choose one of the following procedures depending on whether the existing metadata server is available at the time the change is being performed:

- "To Change the Metadata Server When the Metadata Server Is Available" on page 105
- "To Change the Metadata Server When the Metadata Server Is Unavailable" on page 105

To Change the Metadata Server When the Metadata Server Is Available

This procedure shows how to change the metadata server of a StorageTek QFS shared file system in a StorageTek QFS environment when the metadata server is available.

1. On the existing metadata server, issue the `samsharefs`(1M) `-s` command to declare the new metadata server.

   For example:

```
titan# samsharefs -s tethys sharefs1
```

To Change the Metadata Server When the Metadata Server Is Unavailable

This procedure shows how to change the metadata server of a StorageTek QFS shared file system in a StorageTek QFS environment when the existing metadata server is unavailable.

1. Ensure that the existing metadata server cannot restart without being rebooted.

   Specifically, ensure that the server is powered down, rebooted, halted, or disconnected from the metadata disks. Your goal is to bring down the old metadata server and flush or destroy all buffers (or otherwise ensure that they cannot be rewritten).

Figure 52. shows the key sequence to use from the `kadb` prompt.

Figure 52. Key Sequence for Ensuring that the Metadata Server Cannot Restart from the `kadb` Prompt

```
kadb[1]: :c          # Forces a dump
kadb[1]: $q          # Exits the debugger for prom
```

Figure 53. shows the key sequence to use from the PROM prompt.

Figure 53. Key Sequence for Ensuring that the Metadata Server Cannot Restart from the PROM Prompt

```
{0} > sync            # Forces the buffers out
{0} > boot args       # Discards buffers
```

For *args*, specify arguments for the `boot`(1M) command, such as `-r` or `-v`. For information, see the `boot`(1M) man page.

CAUTION: If the metadata server of a shared file system crashes, it is safe to change the metadata server only *after* rebooting the metadata server or otherwise ensuring that the server cannot issue any I/O prior to being rebooted. Do not use any of the following methods to stop the server because these are likely to corrupt the file system:

- Issuing an L1-A key sequence
- Performing an involuntary failover to another host
- Issuing a go (continue), requesting a dump file, or issuing a `sync` command to the old, down metadata server

Similarly, if the metadata server panics and drops into kernel `adb`(1), do not change the metadata server and then issue `:c` (continue) on the server. This action causes the old metadata server to push stale buffers out to the now active file system.

2. From the new (potential) metadata server, wait for at least the period of the maximum lease time, and then issue the `samsharefs`(1M) command.

   The wait is necessary because you must ensure that all client leases expire before you issue the `samsharefs`(1M) command. From the new metadata server, issue a command such as the following:

```
# samsharefs -R -s tethys sharefs1
```

If you are uncertain as to whether or not the lease time has expired, bring up the samu(1M) N display. For information about samu(1M), see "Using the samu(1M) Operator Utility" on page 131. For information about leases and their durations, see "Using Leases in a StorageTek QFS Shared File System: the rdlease=n, wrlease=n, and aplease=n Options" on page 109.

CAUTION: If you use the -R option to the samsharefs(1M) command on a mounted file system to change the metadata server host, you must first stop, disable, and disconnect the active metadata server. Failure to do so can cause file system corruption.

3. Unmount the file system. (Optional)

   Perform this step only if you want to perform a file system check.

   Use the procedure in "To Unmount a StorageTek QFS Shared File System" on page 93.

4. Issue the samfsck(1M) command to perform a file system check. (Optional)

   Perform this step only if you want to perform a file system check at this time.

   If the metadata server of a StorageTek QFS or StorageTek ASM-QFS shared file system crashes, the server should be rebooted and the file system should be unmounted on all clients before a samfsck(1M) is run. The server and clients preallocate blocks before changing the length of files. The samfsck(1M) command cleans up files that have extra blocks allocated, and these extra blocks might contain data. If such a cleaned-up file is awaiting a size update from the client, the file will be missing those blocks when the client continues. As a result, the file will be missing data, and the missed data will read as zeroes.

Daemons

In a StorageTek QFS shared file system, a sam-fsd daemon is always active. In addition, one sam-sharefsd daemon is active for each mount point configured in the StorageTek QFS shared file system.

When a sam-fsd daemon recognizes a StorageTek QFS shared file system, it starts a shared file system daemon (sam-sharefsd). TCP sockets are used to communicate between the server and client hosts. All clients that connect to the metadata server are validated against the hosts file.

One StorageTek QFS shared file system daemon is started for each StorageTek QFS shared file system shared mount point on each client host. This daemon establishes a connection to the metadata server. The sam-sharedfsd daemon on the metadata server opens a listener socket on the port named sam-qfs. At StorageTek QFS installation time, the sam-qfs entry is added to /etc/services automatically, and this entry should not be removed. The shared file system port is defined in the /etc/inet/services file. The port number installed in the /etc/inet/services file is 7105. Verify that this port does not conflict with another service.

Note: Prior to the StorageTek QFS 4.2 release, one port per file system was required. You can remove these entries from your file.

All metadata operations, block allocation and deallocation, and record locking are performed on the metadata server. The sam-sharefsd daemon does not keep any information. Hence, it can be killed and restarted without causing any consistency problems for the file system.

Mount Options in a StorageTek QFS Shared File System

The StorageTek QFS shared file system can be mounted with several mount options. This chapter describes many options within the context of their roles. Other options, however, are useful only in certain situations. This section describes the mount options that can be used for special purposes.

You can specify most mount options by using the mount(1M) command, by entering them in the /etc/vfstab file, or by entering them in the samfs.cmd(4) file. For example, the following /etc/vfstab file includes mount(1M) options for a StorageTek QFS shared file system:

```
sharefs1 - /sfs samfs - no shared,mh_write
```

You can change some mount options dynamically by using the samu(1M) operator utility. For more information about these options, see “Using the samu(1M) Operator Utility” on page 131.

The following sections summarize the mount options available to you in a StorageTek QFS shared file system. For more information about any of these mount options, see the mount_samfs(1M) man page or see the cross-references mentioned in their descriptions.

Mounting in the Background: the bg Option

The bg mount option specifies that if the first mount operation fails, subsequent attempts at mounting should occur in the background. By default, bg is not in effect, and mount attempts continue in the foreground.

Note: Do *not* use this option if you are mounting a StorageTek QFS shared file system on a Sun Cluster node.

Reattempting a File System Mount: the retry Option

The `retry` mount option specifies the number of times that the system should attempt to mount a file system. The default is 10000.

Declaring a StorageTek QFS Shared File System: the shared Option

The `shared` mount option declares a file system to be a StorageTek QFS shared file system. This option must be specified in the `/etc/vfstab` file in order for the file system to be mounted as a StorageTek QFS shared file system. The presence of this option in a `samfs.cmd`(4) file or on the `mount`(1M) command does not cause an error condition, but it does not mount the file system as a StorageTek QFS shared file system.

Tuning Allocation Sizes: the minallocsz=n and maxallocsz=n Options

The `-o minallocsz=`*n* and `-o maxallocsz=`*n* options to the `mount`(1M) command specify an amount of space, in kilobytes. This is the minimum block allocation size. If a file is growing, the metadata server allocates blocks when an append lease is granted. You can use the `-o minallocsz=`*n* option to specify the initial size of this allocation. The metadata server can increase the size of the block allocation depending on the application's access patterns up to, but not exceeding, the `-o maxallocsz=`*n* option's setting.

You can specify these `mount`(1M) options on the `mount`(1M) command line, in the `/etc/vfstab` file, or in the `samfs.cmd` file.

Using Leases in a StorageTek QFS Shared File System: the rdlease=n, wrlease=n, and aplease=n Options

A *lease* grants a shared host permission to perform an operation on a file for as long as the lease is valid. The metadata server issues leases to each shared host, including itself. The leases are renewed as necessary to permit continued file operations. The possible file operations are as follows:

- A *read lease* enables existing file data to be read.
- A *write lease* enables existing file data to be overwritten.
- An *append lease* enables a file's size to be extended and enables newly allocated blocks to be written.

A shared host can continue to update leases for as long as necessary. The lease is tranparent to the end user. Table 21. shows the mount options that enable you to specify the duration of each lease type.

Table 21. Lease-Related `mount`(1M) Options

Option	Action
`-o rdlease=`*n*	This option specifies the maximum amount of time, in seconds, for the read lease.
`-o wrlease=`*n*	This option specifies the maximum amount of time, in seconds, for the write lease.
`-o aplease=`*n*	This option specifies the maximum amount of time, in seconds, for the append lease.

All three leases enable you to specify an *n* such that $15 \leq n \leq 600$. The default time for each lease is 30 seconds. A file cannot be truncated if a lease is in effect. For more information about setting these leases, see the `mount_samfs`(1M) man page.

If you change the metadata server because the current metadata server is down, you must add the lease time to the change over time because all leases must expire before an alternate metadata server can assume control.

Setting a short lease time causes more traffic between the client hosts and the metadata server because the lease must be renewed after it has expired. For information on lease times in a Sun Cluster environment, see the *StorageTek ASM Installation and Configuration Guide*.

Enabling Multiple Host Reads and Writes: the mh_write Option

By default, in a StorageTek QFS shared file system, multiple hosts can read the same file at the same time, and if no host is writing to that file, I/O can be paged on all hosts. Only one host can append or write to a file at any one time.

The `mh_write` option controls write access to the same file from multiple hosts. If `mh_write` is specified as a mount option on the metadata server host, the StorageTek QFS shared file system enables simultaneous reads and writes to the same file from multiple hosts. If `mh_write` is not specified on the metadata server host, only one host can write to a file at any one time.

By default, `mh_write` is disabled, and only one host has write access to a file at any one time. The length of that time period is determined by the duration of the `wrlease` mount option. If the StorageTek QFS shared file system is mounted on the metadata server with the `mh_write` option enabled, simultaneous reads and writes to the same file can occur from multiple hosts.

Table 22. describes how file access from multiple hosts is affected depending on whether the `mh_write` option is enabled on the metadata server.

Table 22. File Access Based on the `mh_write` Option

`mh_write` **Not Enabled on the Metadata Server**	`mh_write` **Enabled on the Metadata Server**
Multiple reader hosts allowed.	Multiple reader hosts allowed.
Can use paged I/O.	Can use paged I/O.
Only one writer host is allowed. Can use paged I/O. All other hosts wait.	Multiple reader and/or writer hosts allowed. If any writer hosts exist, all I/O is direct.
Only one append host.	Only one append host is allowed.
All other hosts wait.	All other hosts can read and/or write. If any writer hosts exist, all I/O is direct.

The `mh_write` option does not change locking behavior. File locks behave the same whether `mh_write` is in effect or not. The `mh_write` option's effect is as follows:

- When `mh_write` is in effect, all hosts can read from and write to the same file simultaneously.
- When `mh_write` is not in effect, only one host can write to a given file during a given time interval, and no hosts can read from the file during that time interval.

StorageTek QFS shared file system maintains consistency between hosts. The first time that a host executes a read or write system call, it gets a *lease,* which allows it to read or write the file for some period of time. The existence of that lease prevents other hosts without `mh_write` from accessing the file. In particular, the lease can last longer than the duration of the system call that caused its acquisition.

When `mh_write` is not in effect, the StorageTek QFS shared file system should provide near-POSIX behavior for data reads and writes. For metadata, however, access time changes might not be seen immediately on other hosts. Changes to a file are pushed to disk at the end of a write lease, and when a read lease is acquired, the system invalidates any stale cache pages so that the newly written data can be seen.

When `mh_write` is in effect, behavior might be less consistent. When there are simultaneous readers and writers, the StorageTek QFS shared file system switches all hosts accessing the file into direct I/O mode. This means that page-aligned I/O should be visible immediately to other hosts. However, non-page-aligned I/O can result in stale data being visible, or even written to the file, because the normal lease mechanism that prevents this has been disabled.

You should specify the `mh_write` option only when multiple hosts need to write to the same file simultaneously and when applications perform page-aligned I/O. In other cases, there is some risk of data inconsistency because even using `flock`() (which works with `mh_write`) to coordinate between hosts does not guarantee consistency.

For more information about `mh_write`, see the `mount_samfs`(1M) man page.

Setting the Number of Concurrent Threads: the nstreams=n Option

The `nstreams=`*n* mount option sets the number of concurrent threads for the StorageTek QFS shared file system. By default, `nstreams=256`. This means, for example, that under default settings, up to 256 operations can be processed simultaneously, and the 257th operation commences only after an operation has finished. You can adjust the `nstreams=`*n* mount option based on the StorageTek QFS shared file system's activity. For *n*, specify a value such that $76 \leq n \leq 1024$.

Retaining Cached Attributes: the meta_timeo=n Option

The `meta_timeo=`*n* mount option determines how long the system waits between checks on the metadata information. By default, the system refreshes metadata information every 15 seconds. This means, for example, that an `ls`(1) command entered in a StorageTek QFS shared file system with several newly created files might not return information about all the files until 15 seconds had passed. For *n*, specify a value such that $0 \leq n \leq 60$.

Specifying Striped Allocation: the stripe Option

By default, data files in the StorageTek QFS shared file system are allocated using the round-robin file allocation method. To specify that file data be striped across disks, you can specify the `stripe` mount option on the metadata host and all potential metadata hosts. Note that by default, unshared file systems allocate file data using the striped method.

In a round-robin allocation, files are created in a round-robin fashion on each slice or striped group. This causes the maximum performance for one file to be the speed of a slice or striped group. For more information about file allocation methods, see "File System Design" on page 7.

Specifying the Frequency With Which Metadata is Written: the sync_meta=n Option

You can set the `sync_meta=`*n* option to `sync_meta=1` or `sync_meta=0`.

By default, `sync_meta=1` and a StorageTek QFS shared file system writes file metadata to disk every time the metadata changes. This slows data performance, but it ensures data consistency. This is the setting that must be in effect if you want to change the metadata server.

If you set `sync_meta=0`, the StorageTek QFS shared file system writes the metadata to a buffer before writing it to disk. This delayed write delivers higher performance, but it decreases data consistency after an unscheduled machine interruption.

Mount Semantics in a StorageTek QFS Shared File System

The behavior of the StorageTek QFS shared file system is that of an interruptible hard connection. Each client tries repeatedly to communicate with the metadata server, even if the server is unavailable. If the metadata server is not responding, a user can terminate any pending, blocked I/O transmission by pressing Ctrl-C. If the I/O attempt is interrupted, the client persists until the I/O completes.

The system generates the following messages to describe status conditions:

```
SAM-FS: Shared server is not responding.
```

This message is also generated if the client `sam-sharefsd` daemon is not active or if the server `sam-sharefsd` daemon is not active. When the server responds, it generates the following message:

```
SAM-FS: Shared server is responding.
```

If the file system is not mounted on the metadata server, but it is mounted on the client, the system generates the following message:

```
SAM-FS: Shared server is not mounted.
```

When the StorageTek QFS shared file system mounts on the server, it generates the following message:

```
SAM-FS: Shared server is mounted.
```

File Locking in a StorageTek QFS Shared File System

Mandatory locks are not supported. An EACCES error is returned if the mandatory lock is set. Advisory locks are supported. For more information about advisory locks, see the `fcntl`(2) system call.

Performance Considerations

Because the metadata server looks up file names on behalf of all clients, performance can improve if you increase the size of the Solaris directory name lookup cache (DNLC) on the metadata server. This can increase performance when clients are frequently opening a large number of files. Doubling or tripling the size of this cache from its default can be appropriate.

This procedure is documented in the *Solaris Tunable Parameters Reference Manual*. The parameter that controls the size of the directory name lookup cache is `ncsize`.

Troubleshooting a Failed or Hung sammkfs(1M) or mount(1M) Command

The following sections describe what to do when a `sammkfs`(1M) or `mount`(1M) command fails or when a `mount`(1M) command hangs.

The procedures in this section can be performed on client hosts and can also be performed on the server. Commands that can be executed only on the metadata server are preceded with a `server#` prompt.

Recovering a Failed sammkfs(1M) Command

If the `sammkfs`(1M) command returns an error or messages indicating that an unexpected set of devices are to be initialized, you need to perform this procedure. It includes steps for verifying the `mcf`(4) file and for propagating `mcf`(4) file changes to the system.

To Verify the mcf(4) File and Propagate mcf(4) File Changes to the System

1. Use the `sam-fsd`(1M) command to verify the `mcf`(4) file.

   For example:

```
# sam-fsd
```

Examine the output from the `sam-fsd`(1M) command and determine if there are errors that you need to fix.

2. Edit the `mcf`(4) file to resolve any diagnostic issues. (Optional)

   Perform this step if the output from the `sam-fsd`(1M) command indicates that there are errors in the `/etc/opt/SUNWsamfs/mcf` file.

3. Issue the `sam-fsd`(1M) command again to verify the `mcf`(4) file.

   Repeat Step 1, Step 2, and Step 3 of this process until the output from the `sam-fsd`(1M) command indicates that the `mcf`(4) file is correct.

4. Issue the `samd`(1M) `config` command.

   This is needed to propagate `mcf`(4) file changes by informing the `sam-fsd` daemon of the configuration change.

   For example:

```
# samd config
```

Recovering a Failed mount(1M) Command

A `mount`(1M) command can fail for several reasons. This section describes some actions you can take to remedy a mount problem. If the `mount`(1M) command hangs, rather than fails, see "Recovering a Hung mount(1M) Command" on page 120.

Some failed `mount`(1M) behaviors and their remedies are as follows:

- If the mount(1M) command fails with a `Shared server is not mounted` message generated on a client, determine the server host and mount the file system on the metadata server.

- If the mount command fails with a message indicating that there is a mismatch between the file system and the `mcf`(4) file, ensure the following:

  - That the `mcf`(4) file is syntactically valid. For more information, see "To Verify the mcf(4) File and Propagate mcf(4) File Changes to the System" on page 114.

  - That recent changes to the `mcf`(4) file are valid and have been enacted. For more information, see "To Verify the mcf(4) File and Propagate mcf(4) File Changes to the System" on page 114.

  - That the `mcf`(4) file matches the server's `mcf`(4) file with device names or controller numbers adjusted for any differences on the client. You can use the `samfsconfig`(1M) command to diagnose some of these problems. For more information about using the `samfsconfig`(1M) command, see "To Use the samfsconfig(1M) Command" on page 119.

- If the `mount`(1M) command fails for other reasons, use the procedures described in the following sections to verify the system characteristics that must be in place in order for the `mount`(1M) command to be successful. These procedures are as follows:
  - "To Verify that the File System can be Mounted" on page 116
  - "To Use the samfsinfo(1M) and samsharefs(1M) Commands" on page 117
  - "To Use the samfsconfig(1M) Command" on page 119

To Verify that the File System can be Mounted

If this procedure does not expose errors, perform "To Use the samfsinfo(1M) and samsharefs(1M) Commands" on page 117, which can help you verify that the file system has been created and that the shared hosts file is correctly initialized.

The following procedure shows you what to verify if the `mount`(1M) command fails.

1. Ensure that the mount point directory is present.

   There are multiple ways to accomplish this. For example, you can issue the `ls`(1) command in the following format:

```
ls -ld mountpoint
```

   For *mountpoint*, specify the name of the StorageTek QFS shared file system's mount point.

   When you examine the `ls`(1) command's output, make sure that the output shows a directory with access mode 755. In other words, the codes should read `drwxr-xr-x`. Figure 54. shows example output.

Figure 54. Access Mode Values

```
# ls -ld /sharefs1
drwxr-xr-x   2 root     sys          512 Mar 19 10:46 /
sharefs1
```

   If the access is not at this level, enter the following `chmod`(1) command:

```
# chmod 755 mountpoint
```

   For *mountpoint*, specify the name of the StorageTek QFS shared file system's mount point.

2. Ensure that there is an entry for the file system in the `/etc/vfstab` file.

Figure 55. shows an entry for the shared file system named `sharefs1`.

Figure 55. Example `/etc/vfstab` File

```
# File /etc/vfstab
# FS name  FS to fsck  Mnt pt FS type  fsck pass  Mt@boot
Mt params
sharefs1    -          /sharefs1 samfs -          yes
shared,bg
```

Ensure that the `shared` flag is present in the Mount Parameters field of the shared file system's entry in the `/etc/vfstab` file.

3. Ensure that the mount point directory is not shared out for NFS use.

   If the mount point is shared, use the `unshare`(1M) command to unshare it. For example:

```
# unshare mountpoint
```

For *mountpoint*, specify the name of the StorageTek ASM-QFS shared file system's mount point.

To Use the samfsinfo(1M) and samsharefs(1M) Commands

This procedure shows how to analyze the output from these commands.

1. Enter the `samfsinfo`(1M) command on the server.

   Use this command in the following format:

```
samfsinfo filesystem
```

For *filesystem*, specify the name of the StorageTek QFS shared file system as specified in the `mcf`(4) file. Figure 56. shows the `samfsinfo`(1M) command and output.

Figure 56. `samfsinfo`(1M) Command Example

```
titan-server# samfsinfo sharefs1
samfsinfo: filesystem sharefs1 is mounted.
name:     sharefs1       version:     2    shared
time:     Mon Apr 29 15:12:18 2002
count:    3
capacity:      10d84000          DAU:         64
space:         10180400
meta capacity: 009fe200          meta DAU:    16
meta space:    009f6c60
ord  eq   capacity      space   device
```

Figure 56. `samfsinfo`**(1M) Command Example (Continued)**

```
1     11    086c0000    080c39b0    /dev/dsk/
c1t2100002037E9C296d0s6
2     12    086c4000    080bca50    /dev/dsk/
c3t50020F2300005D22d0s6
3     13    086c4000    080a9650    /dev/dsk/
c3t50020F2300006099d0s6
4     14    086c4000    08600000    /dev/dsk/
c3t50020F230000651Cd0s6
```

The output from Figure 56. shows a `shared` keyword in the following line:

```
name:     sharefs1       version:     2     shared
```

Note the list of file system devices, ordinals, and equipment numbers that appear after the following line:

```
ord   eq    capacity       space    device
```

Make sure that these numbers correspond to the devices in the file system's `mcf`(4) entry.

2. Enter the `samsharefs`(1M) command on the server.

   Use this command in the following format:

```
samsharefs -R filesystem
```

For *filesystem*, specify the name of the StorageTek QFS shared file system as specified in the `mcf`(4) file. Figure 57. shows the `samsharefs`(1M) command and output.

Figure 57. `samsharefs`**(1M) Command Example**

```
titan-server# samsharefs -R sharefs1
#
# Host file for family set 'sharefs1'
#
# Version: 3    Generation: 50    Count: 4
# Server = host 0/titan, length = 216
#
titan 173.26.2.129,titan.foo.com 1 - server
tethys 173.26.2.130,tethys.foo.com 2 -
dione dione.foo.com 0 -
mimas mimas.foo.com 0 -
```

The following information pertains to the diagnostic output from the `samfsinfo`(1M) or `samsharefs`(1M) commands.

- If either command issues diagnostics or error messages, resolve them. Ensure that the output from the `samfsinfo`(1M) command includes the `shared` keyword.
- You can execute these commands on alternate server hosts and on client hosts that have no `nodev` devices in the host's `mcf` entry for the file system.

  If the `samfsinfo`(1M) and `samsharefs`(1M) commands do not expose irregularities, perform "To Use the samfsconfig(1M) Command" on page 119.

To Use the samfsconfig(1M) Command

On clients with `nodev` device entries in the `mcf` file for the file system, the entire file system might not be accessible, and the shared hosts file might not be directly accessible. You can use the `samfsconfig`(1M) command to determine whether the shared file system's data partitions are accessible.

1. Issue the `samfsconfig`(1M) command.

   Use this command in the following format:

```
samfsconfig list_of_devices
```

   For *list_of_devices*, specify the list of devices from the file system entry in the `mcf`(4) file. Use a space to separate multiple devices in the list.

   Example 1. Figure 58. shows the `samfsconfig`(1M) command issued on a host that does not have a `nodev` entry in its `mcf` file. Figure 58. shows the `mcf` file for the host `tethys`.

Figure 58. `samfsconfig`**(1M) Command Example Without** `nodev` **Entries**

```
tethys# cat /etc/opt/SUNWsamfs/mcf
sharefs1                          10   ma    sharefs1     on
shared
/dev/dsk/c1t2100002037E9C296d0s6 11   mm    sharefs1     -
/dev/dsk/c3t50020F2300005D22d0s6 12   mr    sharefs1     -
/dev/dsk/c3t50020F2300006099d0s6 13   mr    sharefs1     -
/dev/dsk/c3t50020F230000651Cd0s6 14   mr    sharefs1     -

tethys# samfsconfig /dev/dsk/c1t2100002037E9C296d0s6 /dev/
dsk/c3t50020F2300005D22d0s6 /dev/dsk/c3t50020F2300006099d0s6
/dev/dsk/c3t50020F230000651Cd0s6
#
# Family Set 'sharefs1' Created Mon Apr 29 15:12:18 2002
#
sharefs1                          10    ma    sharefs1   - shared
/dev/dsk/c1t2100002037E9C296d0s6    11     mm    sharefs1   -
```

Figure 58. `samfsconfig`**(1M) Command Example Without** `nodev` **Entries**

```
/dev/dsk/c3t50020F2300005D22d0s6   12    mr    sharefs1   -
/dev/dsk/c3t50020F2300006099d0s6   13    mr    sharefs1   -
/dev/dsk/c3t50020F230000651Cd0s6   14    mr    sharefs1   -
```

Example 2. Figure 59. shows the `samfsconfig`(1M) command being used on a host that has a `nodev` entry in its `mcf` file.

Figure 59. `samfsconfig`**(1M) Command Example With** `nodev` **Entries**

```
dione# cat /etc/opt/SUNWsamfs/mcf
sharefs1                            10    ma    sharefs1   on
shared
nodev                              11    mm    sharefs1   -
/dev/dsk/c4t50020F23000055A8d0s3   12    mr    sharefs1   -
/dev/dsk/c4t50020F23000055A8d0s4   13    mr    sharefs1   -
/dev/dsk/c4t50020F23000055A8d0s5   14    mr    sharefs1   -

dione# samfsconfig /dev/dsk/c4t50020F23000055A8d0s3 /dev/
dsk/c4t50020F23000055A8d0s4 /dev/dsk/c4t50020F23000055A8d0s5
#
# Family Set 'sharefs1' Created Mon Apr 29 15:12:18 2002
#
# Missing slices
# Ordinal 1
# /dev/dsk/c4t50020F23000055A8d0s3    12    mr    sharefs1   -
# /dev/dsk/c4t50020F23000055A8d0s4    13    mr    sharefs1   -
# /dev/dsk/c4t50020F23000055A8d0s5    14    mr    sharefs1   -
```

For examples 1 and 2, verify that the output lists all slices from the file system, other than the metadata (`mm`) devices, as belonging to the file system. This is the case for example 2.

Recovering a Hung mount(1M) Command

If the `mount`(1M) command hangs, follow the procedure in this section. You have a hung `mount`(1M) command if, for example, the `mount`(1M) command fails with a connection error or with a `Server not responding` message that does not resolve itself within 30 seconds.

The most typical remedy for a hung `mount`(1M) command is presented first. If that does not work, perform the subsequent procedures.

To Verify Network Connections

The `netstat`(1M) command verifies that the `sam-sharefsd` daemon's network connections are correctly configured.

1. Become superuser on the metadata server.
2. Type the `samu`(1M) command to invoke the `samu`(1M) operator utility.

   For example:

```
# samu
```

3. Press `P` to access the `Active Services` display.

   Figure 60. shows a `P` display.

Figure 60. `P` Display on the Metadata Server

```
Active Services                                 samu    4.2 09:02:22
Sept 22 2004


Registered services for host 'titan':
    sharedfs.sharefs1
  1 service(s) registered.
```

   Examine the output. In Figure 60., look for a line that contains `sharedfs`.*filesystem-name*. In this example, the line must contain `sharedfs.sharefs1`.

   If no such line appears, you need to verify that both the `sam-fsd` and `sam-sharefsd` daemons have started. Perform the following steps:

   a. Enable daemon tracing in the `defaults.conf` file.

      For information about how to enable tracing, see `defaults.conf`(4) or see Step 2 in "To Examine the sam-sharefsd Trace Log (Optional)" on page 126.

   b. Examine your configuration files, especially `/etc/opt/SUNWsamfs/mcf`.

   c. After you have checked your configuration files and verified that the daemons are active, begin this procedure again.

4. Enter the `samsharefs`(1M) command to check the hosts file.

   Figure 64. shows the `samsharefs`(1M) command and correct output.

Figure 61. `samsharefs`(1M) `-R` Command

```
titan-server# samsharefs -R sharefs1
#
# Host file for family set 'sharefs1'
#
# Version: 3    Generation: 50    Count: 4
# Server = host 0/titan, length = 216
```

Figure 61. `samsharefs`**(1M)** `-R` **Command (Continued)**

```
#
titan 173.26.2.129,titan.foo.com 1 - server
tethys 173.26.2.130,tethys.foo.com 2 -
dione dione.foo.com 0 -
mimas mimas.foo.com 0 -
```

In the output on your system, verify the following:

- That the host name is present in column 1 of the output and that it is designated as the server.
- That the host IP address is present in column 2. If there are multiple IP addresses, make sure that they are all valid.

5. Enter the `netstat`(1M) command on the server.

   Figure 62. shows the `netstat`(1M) command entered on server `titan`.

Figure 62. `netstat`**(1M) Example on the Server**

```
titan-server# netstat -a | grep sam-qfs
      *.sam-qfs *.*             0      0 24576   0 LISTEN
      *.sam-qfs *.*             0      0 24576   0 LISTEN
titan.32834  titan.sam-qfs 32768      0 32768   0 ESTABLISHED
titan.sam-qfs  titan.32891 32768      0 32768   0 ESTABLISHED
titan.sam-qfs tethys.32884 24820      0 24820   0 ESTABLISHED
titan.sam-qfs  dione.35299 24820      0 24820   0 ESTABLISHED
     *.sam-qfs *.*              0      0 24576   0 LISTEN
```

Verify that the output from the `netstat`(1M) command on the server contains the following:

- Three `LISTEN` entries.
- Two `ESTABLISHED` entries for the host.
- One `ESTABLISHED` entry for each client that is configured and running the `sam-fsd` daemon. This example shows `ESTABLISHED` entries for `tethys` and `dione`. There should be one `ESTABLISHED` entry for each client that is configured and running whether or not it is mounted.

6. Enter the `netstat`(1M) command on the client.

   Figure 63. shows the `netstat`(1M) command entered on client `dione`.

Figure 63. `netstat`**(1M) Command on the Client**

```
dione-client# netstat -a | grep sam-qfs
     *.sam-qfs     *.*            0     0 24576       0 LISTEN
     *.sam-qfs     *.*            0     0 24576       0 LISTEN
dione.32831     titan.sam-qfs   24820     0 24820       0 ESTABLISHED
     *.sam-qfs     *.*            0     0 24576       0 LISTEN
```

Verify that the output contains the following:

- Three `LISTEN` entries. All entries are for the `sam-fsd` daemon.
- One `ESTABLISHED` entry.

  If these lines are present, then the network connection is established.

  If an `ESTABLISHED` connection is not reported, go to Step 7.

7. Perform one or more of the following procedures:

- "To Verify that the Client Can Reach the Server (Optional)" on page 123
- "To Verify that the Server Can Reach the Client (Optional)" on page 125
- "To Examine the sam-sharefsd Trace Log (Optional)" on page 126

To Verify that the Client Can Reach the Server (Optional)

Perform these steps if using the procedure described in "To Verify Network Connections" on page 120 did not show an `ESTABLISHED` connection.

1. Use the `samsharefs`(1M) command to verify the hosts file on the server.

   You can issue the `samsharefs`(1M) command on alternate server hosts and client hosts that have no `nodev` devices listed in the host's `mcf`(4) entry for the file system. For this step, use this command in the following format:

```
samsharefs -R filesystem
```

For *filesystem*, specify the name of the StorageTek QFS shared file system as specified in the `mcf`(4) file. Figure 64. shows the `samsharefs`(1M) `-R` command.

Figure 64. `samsharefs`**(1M)** `-R` **Command**

```
titan-server# samsharefs -R sharefs1
#
# Host file for family set 'sharefs1'
#
# Version: 3    Generation: 50    Count: 4
# Server = host 0/titan, length = 216
#
titan 173.26.2.129,titan.xyzco.com 1 - server
tethys 173.26.2.130,tethys.xyzco.com 2 -
dione dione.foo.com 0 -
mimas mimas.foo.com 0 -
```

2. Save this output.

If the steps in this procedure fail, you need this output for use in subsequent procedures.

3. Verify that the output matches expectations.

   If the command fails, verify that the file system was created. In this case it is likely that one of the following has occurred:

   - The `mcf` file was not created properly. You can use the `samfsconfig`(1M) command to verify the correctness of the `mcf` file.
   - The file system was never created.
   - The initial hosts configuration files have not been created. For information about configuring these files, see the procedures earlier in this chapter. The configuration process involves editing the existing `mcf`(4) file, propagating the `mcf`(4) file changes to the rest of the system, and configuring the hosts files.

4. Find the row containing the server's name in the first column.

5. From the client, use the `ping`(1M) command on each entry from the second column of `samsharefs`(1M) output to verify that the server can be reached.

   Use this command in the following format:

```
ping servername
```

   For *servername*, specify the name of the server as shown in the second column of the `samsharefs`(1M) command's output.

   Figure 65. shows output from `ping`(1M).

Figure 65. Using `ping`(1M) on Systems Named in `samsharefs`(1M) Output

```
dione-client# ping 173.26.2.129
ICMP Host Unreachable from gateway dione (131.116.7.218)
for icmp from dione (131.116.7.218) to 173.26.2.129
dione-client# ping titan.xyzco.com
titan.foo.com is alive
```

6. From the client, examine the `hosts.`*filesystem*`.local` file. (Optional)

   Perform this step if the `ping`(1M) command revealed unreachable hosts.

   If there is more than one entry in the second column of `samsharefs`(1M) output, and if some of the entries are not reachable, ensure that only the reachable entries for the entries you want the shared file system to use are present. Also ensure that the necessary entries are present in the `/etc/`

opt/SUNWsamfs/hosts.*filesystem*.local file entry on that host. Ensure that the unreachable hosts are not entered in these places.

If the sam-sharefsd daemon attempts to connect to unreachable server interfaces, there can be substantial delays in its connecting to the server after installation, rebooting, or file system host reconfiguration. This affects metadata server failover operations substantially.

Figure 66. shows the hosts.sharefs1.local file.

Figure 66. Examining the hosts.*filesystem*.local File

```
dione-client# cat /etc/opt/SUNWsamfs/hosts.sharefs1.local
titan       titan.xyzco.com           # no route to
173.26.2.129
tethys      tethys.xyzco.com          # no route to
173.26.2.130
```

7. Enable the correct server interfaces. (Optional)

   If the ping(1M) command revealed that there were no reachable server interfaces, then you need to either configure or initialize the server network interfaces for typical operations, or you must use the samsharefs(1M) command to update the interface names in the hosts file so they match the actual names.

To Verify that the Server Can Reach the Client (Optional)

Perform these steps if the procedure in "To Verify Network Connections" on page 120 did not show an ESTABLISHED connection.

1. Obtain samsharefs(1M) output.

   This can be the output generated in "To Verify that the Client Can Reach the Server (Optional)" on page 123, or you can generate it again using the initial steps in that procedure.

2. Find the row containing the client's name in the first column.

3. On the client, run the hostname(1M) command and ensure that the output matches the name in the first column of samsharefs(1M) output.

   Figure 67. shows the hostname(1M) command and its output.

Figure 67. hostname(1M) Output

```
dione-client# hostname
dione
```

4. Use the ping(1M) command on the server on each entry from the second column to verify that the client can be reached. (Optional)

Perform this step if the `hostname`(1M) command output matched the name in the second column of `samsharefs`(1M) output. Figure 68. shows the `ping`(1M) command and its output.

Figure 68. `ping`**(1M) Output**

```
titan-server# ping dione.xyzco.com
dione.xyzco.com is alive
```

It is not necessary that every entry in column two of Figure 66. be reachable, but all interfaces that you wish any potential server to accept connections from must be present in the column. The server rejects connections from interfaces that are not declared in the shared hosts file.

5. Enable the correct client interfaces. (Optional)

   If the `ping`(1M) command revealed that there were no reachable client interfaces, then either you need to configure or initialize the client network interfaces for typical operations, or you must use the `samsharefs`(1M) command to update the interface names in the hosts file so they match the actual names.

To Examine the sam-sharefsd Trace Log (Optional)

The trace log files keep information generated by the `sam-sharefsd`(1M) daemons during their operation. The trace log files include information about connections attempted, received, denied, refused, and so on, as well as other operations such as host file changes and metadata server changes.

Tracking problems in log files often involves reconciling the order of operations on different hosts by using the log files. If the hosts' clocks are synchronized, log file interpretation is greatly simplified. One of the installation steps directs you to enable the network time daemon, `xntpd`(1M). This synchronizes the clocks of the metadata server and all client hosts during StorageTek QFS shared file system operations.

The trace logs are particularly useful when setting up an initial configuration. The client logs show outgoing connection attempts. The corresponding messages in the server log files are some of the most useful tools for diagnosing network and configuration problems with the StorageTek QFS shared file system. The log files contain diagnostic information for resolving most common problems.

The following procedures can resolve `mount`(1M) problems:

- "To Verify Network Connections" on page 120
- "To Verify that the Client Can Reach the Server (Optional)" on page 123
- "To Verify that the Server Can Reach the Client (Optional)" on page 125

If none of the preceding procedures resolved the problem, perform the steps in this section. You can perform these steps on both the server and the client hosts.

1. Verify the presence of file `/var/opt/SUNWsamfs/trace/sam-sharefsd`.

   If this file is not present, or if it shows no recent modifications, proceed to the next step.

   If the file is present, use `tail`(1) or another command to examine the last few lines in the file. If it shows suspicious conditions, use one or more of the other procedures in this section to investigate the problem.

2. Edit file `/etc/opt/SUNWsamfs/defaults.conf` and add lines to enable `sam-sharefsd` tracing. (Optional)

   Perform this step if Step 1 indicates that file `/var/opt/SUNWsamfs/trace/sam-sharefsd` does not exist or if the file shows no recent modifications.

   a. Copy the example `defaults.conf` file from `/opt/SUNWsamfs/examples/defaults.conf` to `/etc/opt/SUNWsamfs`. (Optional)

      Perform this step if a `defaults.conf` file does not reside in `/etc/opt/SUNWsamfs` at this time. Figure 69. shows this.

Figure 69. Copying the `defaults.conf` File

```
# cd /etc/opt/SUNWsamfs
# cp /opt/SUNWsamfs/examples/defaults.conf .
```

   b. Use `vi`(1) or another editor to edit file `/opt/SUNWsamfs/examples/defaults.conf` and add lines to enable tracing.

      Figure 70. shows the lines to add to the `defaults.conf` file.

Figure 70. Lines to Enable Tracing in `defaults.conf`

```
trace
sam-sharefsd = on
sam-sharefsd.options = all
endtrace
```

   c. Issue the `samd`(1M) `config` command to reconfigure the `sam-fsd`(1M) daemon and cause it to recognize the new `defaults.conf`(4) file.

      For example:

```
# samd config
```

   d. Issue the `sam-fsd`(1M) command to check the configuration files.

Figure 71. shows the output from the `sam-fsd`(1M) command.

Figure 71. Output From the `sam-fsd`(1M) Command

```
# sam-fsd
Trace file controls:
sam-archiverd off
sam-catserverd off
sam-fsd       off
sam-rftd      off
sam-recycler  off
sam-sharefsd  /var/opt/SUNWsamfs/trace/sam-sharefsd
              cust err fatal misc proc date
              size    0    age 0
sam-stagerd   off

Would stop sam-archiverd()
Would stop sam-rftd()
Would stop sam-stagealld()
Would stop sam-stagerd()
Would stop sam-initd()
```

e. Examine the log file in `/var/opt/SUNWsamfs/trace/sam-sharefsd` to check for errors.

```
# more /var/opt/SUNWsamfs/trace/sam-sharefsd
```

3. Examine the last few dozen lines of the trace file for diagnostic information.

   Figure 72. shows a typical `sam-sharefsd` client log file. In this example, the server is `titan`, and the client is `dione`. This file contains normal log entries generated after a package installation, and it finishes with the daemon operating normally on a mounted file system.

Figure 72. Client Trace File

```
dione# tail -18 /var/opt/SUNWsamfs/trace/sam-sharefsd
2004-03-23 16:13:11 shf-shsam2[13835:1]: FS shsam2: Shared file system daemon
started - config only
2004-03-23 16:13:11 shf-shsam2[13835:1]: FS shsam2: Host dione
2004-03-23 16:13:11 shf-shsam2[13835:1]: FS shsam2: Filesystem isn't mounted
2004-03-23 16:13:11 shf-shsam2[13837:1]: FS shsam2: Shared file system daemon
started
2004-03-23 16:13:11 shf-shsam2[13837:1]: FS shsam2: Host dione
2004-03-23 16:13:11 shf-shsam2[13837:1]: FS shsam2: Filesystem isn't mounted
2004-03-23 16:13:11 shf-shsam2[13837:1]: FS shsam2: Kill sam-sharefsd pid 13835
2004-03-23 16:13:12 shf-shsam2[13837:1]: FS shsam2: Killed sam-sharefsd pid 13835
2004-03-23 16:13:12 shf-shsam2[13837:1]: FS shsam2: Host dione; server = titan
2004-03-23 16:13:12 shf-shsam2[13837:1]: FS shsam2: Wakened from AWAIT_WAKEUP
2004-03-23 16:13:14 shf-shsam2[13837:5]: FS shsam2: Set Client (Server titan/3).
```

Figure 72. Client Trace File (Continued)

```
2004-03-23 16:13:14 shf-shsam2[13837:5]: FS shsam2: SetClientSocket dione
(flags=0)
2004-03-23 16:13:14 shf-shsam2[13837:5]: FS shsam2: rdsock dione/0 (buf=6c000).
2004-03-23 16:13:15 shf-shsam2[13837:1]: FS shsam2: Signal 1 received: Hangup
2004-03-23 16:13:15 shf-shsam2[13837:1]: FS shsam2: Wakened from AWAIT_WAKEUP
2004-03-23 16:13:15 shf-shsam2[13837:1]: FS shsam2: mount; flags=18889
2004-03-23 16:18:55 shf-shsam2[13837:1]: FS shsam2: Signal 1 received: Hangup
2004-03-23 16:18:55 shf-shsam2[13837:1]: FS shsam2: Wakened from AWAIT_WAKEUP
```

Using the samu(1M) Operator Utility

6

This chapter shows how to use samu(1M) to control the devices configured within your StorageTek QFS or StorageTek ASM environment. Many samu(1M) displays are useful only for sites using the storage and archive management mechanism. If you are using samu(1M) in a StorageTek QFS environment, some displays do not apply to you.

This chapter contains the following sections:

The operations that you can perform from within samu(1M) can also be performed by using the samcmd(1M) command. For more information about samcmd(1M), see the samcmd(1M) man page.

Overview

The samu(1M) operator utility requires a display terminal that displays a minimum of 24 lines by 80 characters wide. The utility includes the following features:

- Displays that enable you to monitor StorageTek QFS and StorageTek ASM devices and file system activity.
- Commands that enable you to select displays, set display options, control access to and the activity of devices, and take snapshots of display windows.
- Commands that enable you to tune a running StorageTek QFS or StorageTek ASM file system.

The display windows shown in this chapter are representative examples. The exact format and amount of information displayed on your terminal can be different depending on your terminal model and the devices configured in your StorageTek QFS or StorageTek ASM environment.

The following sections describe how to start and stop samu(1M), interact with the utility, access the help windows, and view operator displays.

To Invoke samu(1M)

1. To start samu(1M), type the samu(1M) command from the command line.

   For example:

```
# samu
```

The system starts samu(1M) and shows the help display. This is the default initial display. To view a different samu(1M) display, follow the steps in “To Display a samu(1M) Screen” on page 132.

The samu(1M) utility allows you to select its initial display. For more information about the samu(1M) command line options, see the samu(1M) man page.

Note: samu(1M), like the vi(1) editor, is based on the curses(3CURSES) routine. If you have trouble invoking samu(1M), make sure that your terminal type is defined correctly.

To Display a samu(1M) Screen

The samu(1M) command accepts options on its command line for displaying different samu(1M) screens.

1. Type a colon (:) to bring up the samu(1M) prompt.

   After you type in the colon, the following appears in the lower left:

```
Command:
```

Note: You access the help screen and all the display screens in samu(1M) through this same method. After you type a colon character, the Command prompt appears.

2. Type the letter that corresponds to the display you want to view and press return.

   For example, to view the v display, type a v and press Return after the Command: prompt.

   For a complete list of letters to type and the displays to which they correspond, see “(h) - Help Display” on page 144.

To Stop samu(1M)

1. To exit `samu`(1M), type one of the following:

- `q`
- `:q`

The `samu`(1M) operator utility exits and returns you to the command shell.

Interacting With samu(1M)

Interacting with `samu`(1M) is similar to interacting with the UNIX `vi`(1) editor with respect to paging forward or backward, entering commands, refreshing the display, and quitting the utility.

Each display has its own section in this chapter, and each display section shows the control key sequences you can use to navigate in that display. The `samu`(1M) man page summarizes the control key navigation sequences.

The last line of the display window shows the command and display error messages. If a command error occurs, automatic display refreshing halts until the next operator action.

Entering a Device

Each device included in the StorageTek QFS or StorageTek ASM environment is assigned an Equipment Ordinal (for example, 10) in the `mcf` file. Many `samu`(1M) commands reference a specific device.

Example. The syntax for the `:off` command is as follows:

```
:off eq
```

For *eq*, type the Equipment Ordinal for the device you are trying to address.

Getting Online Help

When you start `samu`(1M), the system automatically displays the first help screen. This help screen differs depending on whether you have a StorageTek QFS or StorageTek ASM file system.

For more information about the help (`h`) display, see "(h) - Help Display" on page 144.

To Access Online Help From a Display Screen

1. Type `:h`

   To move forward or backward from one screen to the next, type the following key sequence:

- Press Ctrl-f to page the display forward.
- Press Ctrl-b to page the display backward to previous pages.

  You can return to the help display at any time by pressing the h key.

Operator Displays

You can view the `samu`(1M) operator displays by pressing the key corresponding to each display. The lowercase keys `a` through `w` display operational information.

Note: The uppercase `samu`(1M) displays (`A`, `C`, `F`, `I`, `J`, `L`, `M`, `N`, `P`, `R`, `S`, `T`, and `U`) are designed to be used at a customer site only with the assistance of a member of the technical support staff.

This chapter does not describe these uppercase displays as thoroughly as the lowercase displays.

For displays that overflow the screen area, the word `more` appears on the bottom of the screen display, indicating that the display contains additional information. Figure 73. contains the word `more`, indicating that more information appears on subsequent screens.

Figure 73. `samu`**(1M) Screen That Indicates More Text Can Be Obtained**

```
xb54  54  exb8505  pt03  0  yes  2  0  on
lt55  55  dlt2000  pt02  1  yes  4  0  on  ml65
hp56  56  hpc1716  pt01  1  yes  3  0  on  hp70
hp57  57  hpc1716  pt01  1  yes  4  0  on  hp70
more
```

If `samu`(1M) prompts you to enter a device, enter its associated Equipment Ordinal. The configuration display (`c`) shows Equipment Ordinals for all removable media devices. To control all displays, use the control keys listed for the display.

The following sections describe the operator displays in alphabetical order. Examples are provided, and when necessary, displays are followed by a table describing the fields displayed.

(a) - Archiver Status Display

The `a` display shows the archiver status.

You can invoke this display differently, depending on what you need to view, as follows:

- To display an archiver status summary, which shows the status of the archiver on a per-file-system basis, type the command with the following format:

```
Command:a
```

- To display archiving details for a specific file system, type the command with the following format:

```
Command:a filesystem
```

For *filesystem*, specify the name of a file system

Navigation

Table 23. shows the control keys you can use in the `a` display.

Table 23. Control Keys for the `a` Display

Key	Function
Ctrl-b	Previous file system
Ctrl-f	Next file system
Ctrl-d	Page *arcopies* forward (bottom portion)
Ctrl-u	Page *arcopies* backward (bottom portion)

Table 24. shows the control keys you can use in the `:a` *filesystem* display.

Table 24. Control Keys for the `:a` *filesystem* Display

Key	Function
Ctrl-b	Previous file system
Ctrl-f	Next file system

Sample Display

Figure 74. shows activity and statistics for a single file system in the summary display.

Figure 74. `samu`**(1M)** `a` **Display**

```
Archiver status                        samu 4.2 07:44:02 Sept 8
2004
License: License never expires.

sam-archiverd:  Waiting for resources



sam-arfind:  samfs1 mounted at /sam1
Waiting until 2004-05-08 07:54:02 to scan .inodes

sam-arfind:  samfs2 mounted at /sam2
Waiting until 2004-05-08 07:52:57 to scan .inodes

sam-arfind:  qfs1 mounted at /qfs1
Waiting until 2004-05-08 07:44:33 to scan .inodes

sam-arfind:  qfs2 mounted at /qfs2
Waiting until 2004-05-08 07:53:21 to scan .inodes

sam-arfind:  qfs3 mounted at /qfs3
Waiting until 2004-05-08 07:44:11 to scan .inodes

sam-arfind:  qfs4 mounted at /qfs4
Waiting until 2004-05-08 07:53:35 to scan .inodes

sam-arfind:  shareqfs1 mounted at /shareqfs1
Shared file system client.  Cannot archive.

sam-arfind:  shareqfs2 mounted at /shareqfs2
Shared file system client.  Cannot archive.


sam-arcopy:  qfs4.arset5.1.83 dt.DAT001
Waiting for volume dt.DAT001
```

Field Descriptions

Table 25. shows the fields in the detail display.

Table 25. `samu`**(1M)** `a` **Display Field Descriptions**

Field	Description
`samfs1 mounted at`	Mount point.
`regular files`	Number of regular files and their total size.
`offline files`	Number of offline files and their total size.
`archdone files`	Number of `archdone` files and size. Indicates that the archiver has completed processing and can perform no further processing for `archdone` files. Files marked as `archdone` have been processed for archiving but have not necessarily been archived.
`copy1`	Number of files and total size for archive copy 1.
`copy2`	Number of files and total size for archive copy 2.
`copy3`	Number of files and total size for archive copy 3.
`copy4`	Number of files and total size for archive copy 4.
`Directories`	Number of directories and total size.
`sleeping until`	Indicates when archiver runs again.

(c) - Device Configuration Display

The `c` display shows your configuration's connectivity. It lists all device names and Equipment Ordinals.

To invoke the device configuration display, type the command with the following format:

```
Command:c
```

Navigation

Table 26. shows the control keys you can use in this display.

Table 26. Control Keys for the `c` Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-u	Half-page backward

Sample Display

Figure 75. shows the device configuration display.

Figure 75. `samu`(1M) `c` Display

```
Device configuration:                     samu     4.2 07:48:11
Sept 8 2004
License: License never expires.

ty   eq state   device_name                        fs
family_set
sk  100 on      /etc/opt/SUNWsamfs/dcstkconf      100 dcL700
tp  120 off     /dev/rmt/1cbn                     100 dcL700
sg  130 on      /dev/rmt/4cbn                     100 dcL700
sg  140 on      /dev/rmt/5cbn                     100 dcL700
tp  150 off     /dev/rmt/3cbn                     100 dcL700
hy  151 on      historian                         151
```

Field Descriptions

Table 27. shows the field descriptions for this display.

Table 27. `samu`(1M) `c` Display Field Descriptions

Field	Description
`ty`	Device type.
`eq`	Equipment Ordinal of the device.
`state`	Current operating state of the device. Valid device states are as follows: • `on`—The device is available for access. • `ro`—The device is available for read-only access. • `off`—The device is not available for access. • `down`—The device is available only for maintenance access. • `idle`—The device is not available for new connections. Operations in progress continue until completion.
`device_name`	Path to the device.
`fs`	Family Set Equipment Ordinal.
`family_set`	Name of the storage Family Set or library to which the device belongs.

(C) - Memory Display

The `C` display shows the content of a specified memory address. To show the content at an address, enter the address in hexadecimal.

To invoke this display, type the following command:

```
Command:C hex_address
```

For *hex_address*, specify the address of a memory location in hexadecimal. For example:

```
Command:C 0x1044a998
```

This display is designed for debugging. It is intended to be used only with the assistance of a StorageTek support staff person.

Sample Display

Figure 76. shows the memory display. The output has been truncated for inclusion in this manual.

Figure 76. samu**(1M)** C **Display**

```
Memory    base: 0x1234567                samu 4.2 07:52:25 Sept 8 2004
License: License never expires.
00000000  80921000 137ff801 edd05e21 40853420  ......x.mP^!@.4
00000010  00a00420 018528b0 01a604e0 02840640  . . ..(0.&.‘...@
00000020  02d030a1 a0853420 0080a0a0 100a6fff  .P0! .4 ..  ..o.
00000030  f6921000 13c65e23 582d0000 0ba01020  v....F^#X-... .
00000040  00c45e20 48c608e0 2fd05e21 40920080  .D^ HF.‘/P^!@...
00000050  037ff801 fa941000 16c45e20 48a600a0  ..x.z....D^ H&.
00000060  80921000 137ff801 d5d05e21 40853420  ......x.UP^!@.4
00000070  00a00420 018528b0 01a604e0 02840640  . . ..(0.&.‘...@

00000080  02d030a1 c0853420 0080a0a0 100a6fff  .P0!@.4 ..  ..o.
00000090  f6921000 13c65e23 58a01020 00c45e20  v....F^#X . .D^
000000a0  48c608e0 2fd05e21 40920080 037ff801  HF.‘/P^!@.....x.
000000b0  e39405a2 00c45e20 48a600a0 80921000  c..".D^ H&. ....
000000c0  137ff801 bed05e21 40853420 00a00420  ..x.>P^!@.4 . .
000000d0  018528b0 01a604e0 02840640 02d030a1  ..(0.&.‘...@.P0!
000000e0  e0853420 0080a0a0 100a6fff f6921000  ‘.4 ..  ..o.v...
000000f0  13c65e23 58a01020 00c45e20 48c608e0  .F^#X . .D^ HF.‘

00000100  02d05e21 40920080 037ff801 cc941020  .P^!@.....x.L..
00000110  10c45e20 48a600a0 80921000 137ff801  .D^ H&. ......x.
00000120  a7d05e21 40853420 00a00420 018528b0  ’P^!@.4 . . ..(0
00000130  01a604e0 02840640 02d030a2 00853420  .&.‘...@.P0"..4
00000140  0080a0a0 400a6fff f6921000 13c65e23  ..  @.o.v....F^#
00000150  58a01020 00c45e20 48c608e0 02d05e21  X . .D^ HF.‘.P^!
00000160  40920080 037ff801 b5941020 20c45e20  @.....x.5..  D^
00000170  48a600a0 80921000 137ff801 90d05e21  H&. ......x..P^!

00000180  40853420 00a00420 018528b0 01a604e0  @.4 . . ..(0.&.‘
00000190  02840640 02d030a2 80853420 0080a0a0  ...@.P0"..4 ..
000001a0  400a6fff f6921000 13c65e23 58a01020  @.o.v....F^#X .
000001b0  00c45e20 48c608e0 02d05e21 40920080  .D^ HF.‘.P^!@...
000001c0  037ff801 9e941020 30c45e20 48a600a0  ..x.... 0D^ H&.
000001d0  80921000 137ff801 79d05e21 40853420  ......x.yP^!@.4
000001e0  00a00420 018528b0 01a604e0 02840640  . . ..(0.&.‘...@
000001f0  02d030a3 00853420 0080a0a0 400a6fff  .P0#..4 ..  @.o.
```

(d) - Daemon Trace Controls Display

The d display shows the events being traced as specified in the defaults.conf file. For more information about enabling trace files, see the defaults.conf(4) man page.

To invoke this display, type the following command:

```
Command:d
```

Sample Display

Figure 77. shows trace file information. It includes information about the daemons being traced, the paths to the trace files, the events being traced, and information about the size and age of the trace files.

Figure 77. `samu`**(1M)** `d` **Display**

```
Daemon trace controls                   samu 4.2 07:56:38 Sept
8 2004
License: License never expires.

sam-amld       /var/opt/SUNWsamfs/trace/sam-amld
               cust err fatal misc proc debug date
               size    0    age 0

sam-archiverd  /var/opt/SUNWsamfs/trace/sam-archiverd
               cust err fatal misc proc debug date
               size    0    age 0

sam-catserverd /var/opt/SUNWsamfs/trace/sam-catserverd
               cust err fatal misc proc debug date
               size    0    age 0

sam-fsd        /var/opt/SUNWsamfs/trace/sam-fsd
               cust err fatal misc proc debug date
               size    0    age 0

sam-rftd       /var/opt/SUNWsamfs/trace/sam-rftd
               cust err fatal misc proc debug date
               size    0    age 0

sam-recycler   /var/opt/SUNWsamfs/trace/sam-recycler
               cust err fatal misc proc debug date
               size    0    age 0

sam-sharefsd   /var/opt/SUNWsamfs/trace/sam-sharefsd
               cust err fatal misc proc debug date
               size    0    age 0

sam-stagerd    /var/opt/SUNWsamfs/trace/sam-stagerd
               cust err fatal misc proc debug date
               size    0    age 0

sam-serverd    /var/opt/SUNWsamfs/trace/sam-serverd
               cust err fatal misc proc debug date
               size    0    age 0
```

Figure 77. `samu`**(1M)** `d` **Display (Continued)**

```
sam-clientd    /var/opt/SUNWsamfs/trace/sam-clientd
               cust err fatal misc proc debug date
               size    0    age 0

sam-mgmt       /var/opt/SUNWsamfs/trace/sam-mgmt
               cust err fatal misc proc debug date
               size    0    age 0
```

(f) - File Systems Display

The `f` display shows the components of your StorageTek QFS or StorageTek ASM file systems.

To invoke this display, type the following command:

```
Command:f
```

Sample Display

Figure 78. shows the file systems display.

Figure 78. `samu`**(1M)** `f` **Display**

```
File systems                                         samu 4.2 08:11:24 Sept 8
2004
License: License never expires.

ty      eq state          device_name      status high low mountpoint
server
ms      10    on                samfs1  m----2----d 90% 70% /sam1
 md     11    on     /dev/dsk/c5t8d0s3
 md     12    on     /dev/dsk/c5t8d0s4
 md     13    on     /dev/dsk/c5t8d0s5
 md     14    on     /dev/dsk/c5t8d0s6
 md     15    on     /dev/dsk/c5t8d0s7
ms      20    on                samfs2  m----2----d 90% 70% /sam2
 md     21    on     /dev/dsk/c5t9d0s3
 md     22    on     /dev/dsk/c5t9d0s4
 md     23    on     /dev/dsk/c5t9d0s5
 md     24    on     /dev/dsk/c5t9d0s6
 md     25    on     /dev/dsk/c5t9d0s7
ma      30    on                  qfs1  m----2----d 90% 70% /qfs1
 mm     31    on    /dev/dsk/c5t10d0s0
 md     32    on    /dev/dsk/c5t10d0s1
ma      40    on                  qfs2  m----2----d 90% 70% /qfs2
 mm     41    on    /dev/dsk/c5t11d0s0
 md     42    on    /dev/dsk/c5t11d0s1
ma      50    on                  qfs3  m----2---r- 90% 70% /qfs3
```

Figure 78. `samu`**(1M)** `f` **Display (Continued)**

```
 mm     51     on    /dev/dsk/c5t12d0s0
 mr     52     on    /dev/dsk/c5t12d0s1
ma      60     on                 qfs4  m----2---r- 90% 70% /qfs4
 mm     61     on    /dev/dsk/c5t13d0s0
 mr     62     on    /dev/dsk/c5t13d0s1
ma     100    on            shareqfs1  m----2c--r- 80% 70% /shareqfs1 spade
 mm    101     on /dev/dsk/c6t50020F2300004655d0s0
 mr    102     on /dev/dsk/c6t50020F2300004655d0s1
ma     110    on            shareqfs2  m----2c--r- 80% 70% /shareqfs2 spade
 mm    111     on /dev/dsk/c6t50020F2300004655d0s6
 mr    112     on /dev/dsk/c6t50020F2300004655d0s7
```

Field Descriptions

Table 28. shows the field descriptions for this display.

Table 28. `samu`**(1M)** `f` **Display Field Descriptions**

Field	Description
`ty`	Device type.
`eq`	Equipment Ordinal of the device.
`state`	Current operating state of the device. Valid device states are as follows: • `on`—The device is available for access. • `ro`—The device is available for read-only access. • `off`—The device is not available for access. • `down`—The device is available only for maintenance access. • `idle`—The device is not available for new operations. Operations in progress continue until completion.
`device_name`	File system name or path to the device.
`status`	Device status. For a description of status codes, see “Operator Display Status Codes” on page 175.
`high`	High disk usage threshold percentage.
`low`	Low disk usage threshold percentage.
`mountpoint`	Mount point of the file system.
`server`	Name of the host system upon which the file system is mounted.

(F) - Optical Disk Label Display

The `F` display shows the label on an optical disk.

To invoke this display, type the following command:

```
Command:F
```

This display is designed for debugging. It is intended to be used only with the assistance of a StorageTek support staff person.

(h) - Help Display

The `h` display shows a summary of the `samu`(1M) displays available. By default, this is the first display that the system presents when you enter the `samu`(1M) command at the command line.

To invoke this display, type the following command:

```
Command:h
```

Navigation

Table 29. shows the control keys you can use in this display.

Table 29. Control Keys for the `h` Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Page forward (top portion)
Ctrl-f	Page forward
Ctrl-u	Page backward (top portion)
Ctrl-k	Toggle path display

Sample Display

There are several pages of help screens, but this manual shows only the first. Subsequent help screens show `samu`(1M) commands.

Figure 79. shows the initial help screen for the StorageTek ASM file system. On a StorageTek QFS file system, not all of the displays appear in the initial help screen. For example, the removable media displays are not available if

you are running a StorageTek QFS system. If you are using StorageTek ASM-QFS software, the help screen appears as shown in Figure 79..

Figure 79. `samu`**(1M) Initial Help Screen for a StorageTek ASM System**

```
Help information            page 1/15   samu 4.2          08:18:13 Sept 8
2004
License: License never expires.

Displays:

    a   Archiver status                    w       Pending stage queue
    c   Device configuration               C       Memory
    d   Daemon trace controls              F       Optical disk label
    f   File systems                       I       Inode
    h   Help information                   J       Preview shared memory
    l   License information                K       Kernel statistics
    m   Mass storage status                L       Shared memory tables
    n   Staging status                     M       Shared memory
    o   Optical disk status                N       File system parameters
    p   Removable media load requests      P       Active Services
    r   Removable media                    R       SAM-Remote
    s   Device status                      S       Sector data
    t   Tape drive status                  T       SCSI sense data
    u   Staging queue                      U       Device table
    v   Robot catalog


    more (ctrl-f)
```

(I) - Inode Display

The `I` display shows the content of inodes.

You can invoke this display differently, depending on what you need to view, as follows:

- To display inodes for an entire file system, type the command with the following format:

  `Command:I` *filesystem*

  For *filesystem*, specify the name of a file system

- To display a specific inode, type the command with the following format:

  `Command:I` *inode_number*

  For *inode_number*, specify the inode number in either hexadecimal or decimal.

Navigation

Table 30. shows the control keys you can use in this display.

Table 30. Control Keys for the I Display

Key	Function
Ctrl-b	Previous inode
Ctrl-f	Next inode
Ctrl-k	Advance display format

This display is designed for debugging. It is intended to be used only with the assistance of a StorageTek support staff person.

Sample Display

Figure 80. shows the inode display.

Figure 80. `samu`**(1M)** `I` **Display**

```
Inode     0x1 (1) format: file          samu 4.2          08:27:14 Sept 8
2004
License: License never expires.                       incore: y

00008100 mode       -r---------          409cdf57 access_time
00000001 ino        (1)                  1d32ea20
00000001 gen        (1)                  4096b499 modify_time
00000002 parent.ino (2)                  02588660
00000002 parent.gen (2)                  4096b499 change_time
00000000 size_u                          02588660
000c0000 size_l     (786432)             4096b443 creation_time
01000000 rm:media/flags                  409a8a7c attribute_time
00000000 rm:file_offset                  409c0ce6 residence_time
00000000 rm:mau                          00000000 unit/cs/arch/flg
00000000 rm:position                     00000000 ar_flags
00000000 ext_attrs  --------             00000000 stripe/stride/sg
00000000 ext.ino    (0)                  00000000 media   -- --
00000000 ext.gen    (0)                  00000000 media   -- --
00000000 uid        root                 00000000 psize      (0)
00000000 gid        root                 000000c0 blocks     (192)
00000001 nlink      (1)                  00000600 free_ino   (1536)
00011840 status -n-------  ----- -- --

Extents (4k displayed as 1k):
00_ 000000d0.00 000000e0.00 000000f0.00 00000100.00 00000110.00
00000120.00
```

Figure 80. `samu`**(1M)** `I` **Display (Continued)**

```
06_ 00000130.00 00000140.00 00000150.00 00000160.00 00000170.00
00000180.00
12_ 00000190.00 000001a0.00 000001b0.00 000001c0.00 00000630.00
00000000.00
18_ 00000000.00
```

(J) - Preview Shared Memory Display

The `J` display shows the shared memory segment for the preview queue.

To invoke this display, type the following command:

```
Command:J
```

Navigation

Table 31. shows the control keys you can use in this display.

Table 31. Control Keys for the `J` **Display**

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-u	Half-page backward

This display is designed for debugging. It is intended to be used only with the assistance of a StorageTek support staff person.

Sample Display

Figure 81. shows the preview shared memory display. This output has been truncated for inclusion in this manual.

Figure 81. `samu`**(1M)** `J` **Display**

```
Preview shared memory   size: 155648    samu 4.2         08:30:05 Sept 8 2004
License: License never expires.
00000000  00040000 00014d58 00000000 00000000  ......MX........
00000010  00000000 00000000 73616d66 73202d20  ........samfs -
00000020  70726576 69657720 6d656d6f 72792073  preview memory s
00000030  65676d65 6e740000 00026000 00000000  egment....‘.....
00000040  00025fff 00000000 00040000 00014d58  .._..........MX
00000050  00000000 00000000 00000000 00000000  ................
00000060  0000d9e0 00000064 00000000 000001b8  ..Y‘...d.......8
00000070  3f800000 447a0000 0000d820 00000008  ?...Dz....X ....
```

(K) - Kernel Statistics Display

The K display shows kernel statistics, such as the number of inodes currently in memory.

To invoke this display, type the following command:

```
Command:K
```

Navigation

Table 32. shows the control keys you can use in this display.

Table 32. Control Keys for the K Display

Key	Function
Ctrl-b	Page backward
Ctrl-f	Page forward

This display is designed for debugging. It is intended to be used only with the assistance of a StorageTek support staff person.

Sample Display

Figure 82. shows the kernel statistics display.

Figure 82. samu**(1M)** K **Display**

```
Kernel statistics                        samu 4.2         08:33:19 Sept 8 2004
License: License never expires.

module: sam-qfs  name: general instance: 0 class: fs
version                       4.2.sam-qfs, gumball 2004-05-07 12:12:04
configured file systems       8
mounted file systems          8
nhino                         16384
ninodes                       129526
inocount                      129527
inofree                       128577
```

(l) - License Display

The l display shows shows the licenses and expiration dates for StorageTek QFS and StorageTek ASM software.

To invoke this display, type the following command:

```
Command:l
```

Sample Display

Figure 83. shows an example of a license display.

Figure 83. `samu`**(1M)** `l` **Display**

```
License information                           samu     4.2    08:36:27 Sept 8
2004
License: License never expires.

hostid = 80e69e6e

License never expires
Remote sam server feature enabled
Remote sam client feature enabled
Migration toolkit feature enabled
Fast file system feature enabled
Data base feature enabled
Foreign tape support enabled
Segment feature enabled
Shared filesystem support enabled
SAN API support enabled

Robot type STK ACSLS Library is present and licensed
    350 sg slots present and licensed
```

The sample display shows license information for a StorageTek ASM file system. The license information is derived from the license keys in the following file:

```
/etc/opt/SUNWsamfs/LICENSE.4.2
```

This display shows the following information for the system:

- Expiration information
- Host ID
- StorageTek QFS and StorageTek ASM products and features enabled
- Equipment/media combinations

(L) - Shared Memory Tables

The `L` display shows the location of the shared memory tables. It also shows some system defaults that are kept in shared memory.

To invoke this display, type the following command:

```
Command:L
```

This display is designed for debugging. It is intended to be used only with the assistance of a StorageTek support staff person.

Sample Display

Figure 84. shows the shared memory tables.

Figure 84. `samu`**(1M)** `L` **Display**

```
Shared memory tables                        samu 4.2 08:38:31 May  8 2004
License: License never expires.

shm ptr tbl:                                defaults:
size             12000 (73728)              optical          mo
left             44c8 (17608)               tape             lt
scanner pid      1861                       timeout          600
fifo path        01b0 /var/opt/SUNWsamfs/previews            100
dev_table        01cc                       stages           1000
first_dev        0450                       log_facility     184
scan_mess        cf50                       dio minfilesize  100
preview_shmid    1                          label barcode    FALSE
flags            0x20000000                 barcodes low     FALSE
preview stages   55776                      export unavail   FALSE
preview avail    100                        attended         TRUE
preview count    0                          start rpc        FALSE
preview sequence  445
age factor          1                       vsn factor       1000
fs tbl ptr 0xd820                           fs count  8
fseq  10 samfs1 state 0      0      0      0      0
fseq  20 samfs2 state 0      0      0      0      0
fseq  30 qfs1 state 0      0      0      0      0
fseq  40 qfs2 state 0      0      0      0      0
fseq  50 qfs3 state 0      0      0      0      0
fseq  60 qfs4 state 0      0      0      0      0
fseq  100 shareqfs1 state 0      0      0      0      0
fseq  110 shareqfs2 state 0      0      0      0      0
```

(m) - Mass Storage Status Display

The `m` display shows the status of mass storage file systems and their member drives. This display shows only mounted file systems.

To invoke this display, type the following command:

```
Command:m
```

Sample Display

Figure 85. shows the m display. Member drives are indented one space and appear directly below the file system to which they belong.

Figure 85. samu**(1M)** m **Display**

```
Mass storage status                        samu 4.2          08:41:11 Sept 8 2004
License: License never expires.

ty       eq  status        use state ord  capacity      free     ra   part high low
ms       10  m----2----d    1% on          68.354G   68.343G     1M     16  90% 70%
 md       11                1% on       0  13.669G   13.666G
 md       12                1% on       1  13.669G   13.667G
 md       13                1% on       2  13.669G   13.667G
 md       14                1% on       3  13.674G   13.672G
 md       15                1% on       4  13.674G   13.672G
ms       20  m----2----d    1% on          68.354G   68.344G     1M     16  90% 70%
 md       21                1% on       0  13.669G   13.667G
 md       22                1% on       1  13.669G   13.667G
 md       23                1% on       2  13.669G   13.667G
 md       24                1% on       3  13.674G   13.672G
 md       25                1% on       4  13.674G   13.672G
ma       30  m----2----d    4% on          64.351G   61.917G     1M     16  90% 70%
 mm       31                1% on       0   4.003G    3.988G [8363840 inodes]
 md       32                4% on       1  64.351G   61.917G
ma       40  m----2----d    1% on          64.351G   64.333G     1M     16  90% 70%
 mm       41                1% on       0   4.003G    3.997G [8382784 inodes]
 md       42                1% on       1  64.351G   64.333G
ma       50  m----2---r-    1% on          64.351G   64.333G     1M     16  90% 70%
 mm       51                1% on       0   4.003G    3.997G [8382784 inodes]
 mr       52                1% on       1  64.351G   64.333G
ma       60  m----2---r-    1% on          64.351G   64.331G     1M     16  90% 70%
 mm       61                1% on       0   4.003G    3.997G [8382784 inodes]
 mr       62                1% on       1  64.351G   64.331G
ma      100  m----2c--r-    2% on         270.672G  265.105G     1M     16  80% 70%
 mm      101                1% on       0   2.000G    1.988G [4168992 inodes]
 mr      102                2% on       1 270.672G  265.469G
ma      110  m----2c--r-    3% on         270.656G  263.382G     1M     16  80% 70%
 mm      111                1% on       0   2.000G    1.987G [4167616 inodes]
 mr      112                2% on       1 270.656G  264.736G
```

Field Descriptions

Table 33. shows the field descriptions for this display.

Table 33. samu**(1M)** m **Display Field Descriptions**

Field	Description
ty	Device type.
eq	Equipment Ordinal of the mass storage device.
status	Device status. For a description of status codes, see “Operator Display Status Codes” on page 175.

Table 33. `samu`(1M) `m` Display Field Descriptions (Continued)

Field	Description
`use`	Percentage of disk space in use.
`state`	Current operating state of the mass storage device.
`ord`	Ordinal number of the disk device within the storage Family S et.
`capacity`	Number of 1024-byte blocks of usable space on the disk.
`free`	Number of 1024-byte blocks of disk space available.
`ra`	Readahead size in kilobytes.
`part`	Partial stage size in kilobytes.
`high`	High disk usage threshold percentage.
`low`	Low disk usage threshold percentage.

(M) - Shared Memory Display

The `M` display shows the raw shared memory segment in hexadecimal. This is a device table.

To invoke this display, type the following command:

```
Command:M
```

Navigation

Table 34. shows the control keys you can use in this display.

Table 34. Control Keys for the `M` Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-u	Half-page backward

This display is designed for debugging. It is intended to be used only with the assistance of a StorageTek support staff person.

Sample Display

Figure 86. shows the shared memory display. The output has been truncated for inclusion in this manual.

Figure 86. `samu`**(1M)** `M` **Display**

```
Shared memory        size: 73728          samu 4.2          08:43:20 May  8
2004
License: License never expires.
00000000  00040000 00014d58 00000000 00000000  ......MX........
00000010  00000000 00000000 73616d66 73202d20  ........samfs -
00000020  73686172 6564206d 656d6f72 79207365  shared memory se
00000030  676d656e 74000000 00012000 000044c8  gment..... ...DH
00000040  0000dd20 00000000 00000742 00000745  ..] .......B...E
00000050  00000001 00000000 00000000 c0000000  ...........@...
00000060  00000001 0001534d 00000000 00000000  ......SM........
00000070  00000000 00000000 00000000 00000000  ...............

00000080  00000000 00000000 00000000 00000000  ................
00000090  20000000 000001b0 000001cc 00000450   ......0...L...P
000000a0  0000cf50 00000001 00000001 4c696365  ..OP........Lice
000000b0  6e73653a 204c6963 656e7365 206e6576  nse: License nev
000000c0  65722065 78706972 65732e00 00000000  er expires......
000000d0  00000000 00000000 00000000 00000000  ................
000000e0  00000000 00000000 00000000 00000000  ................
000000f0  00000000 00000000 00000000 00000000  ...............

00000100  00000000 00000000 00000000 00000000  ...............
00000110  00000000 00000000 00000000 00000000  ...............
00000120  00000000 00000000 00000000 00000000  ...............
00000130  00000000 00000000 00000000 00000000  ...............
00000140  00000000 00000000 00000000 00000000  ...............
00000150  00000000 00000000 00000000 00000000  ...............
00000160  00000000 00000000 00000000 00000000  ...............
00000170  00000000 00000000 00000000 00000000  ...............

00000180  00000000 00000000 00000000 00000000  ...............
00000190  00000000 00000000 00000000 00000000  ...............
000001a0  00000000 00000000 00000000 00000000  ...............
000001b0  2f766172 2f6f7074 2f53554e 5773616d  /var/opt/SUNWsam
000001c0  66732f61 6d6c6400 00000000 00040000  fs/amld.........
000001d0  00014d58 00000000 00000000 00000000  ..MX............
000001e0  00000000 00000097 00000000 00000000  ................
000001f0  00000000 00000000 00000000 00000000  ................
```

(n) - Staging Status Display

The `n` display shows the status of the stager for all media. It displays a list of outstanding stage requests.

You can invoke this display differently, depending on what you need to view, as follows:

- To display the staging status for all staging activity, type the command with the following format:

```
Command:n
```

- To display the staging status for a specific media type, type the command with the following format:

```
Command:n mt
```

For *mt*, specify one of the media types shown in the `mcf`(4) man page.

Sample Display

Figure 87. shows the staging status display.

Figure 87. `samu`**(1M)** `n` **Display**

```
Staging status                                    samu 4.2          08:47:16 May  8
2004
License: License never expires.

Log output to: /var/opt/SUNWsamfs/stager/log

Stage request: dt.DAT001
Loading VSN DAT001


Staging queues
ty pid    user         status     wait files vsn
dt 16097  root         active     0:00     12 DAT001
```

(N) - File System Parameters Display

The `N` display shows all mount point parameters, the superblock version, and other file system information.

To invoke this display, type the following command:

```
Command:N
```

Navigation

Table 35. shows the control keys you can use in this display.

Table 35. Control Keys for the `N` Display

Key	Function
Ctrl-b	Previous file system
Ctrl-d	Page partitions forward
Ctrl-f	Next file system
Ctrl-i	Detailed status interpretations
Ctrl-u	Page partitions backward

This display is designed for debugging. It is intended to be used only with the assistance of a StorageTek support staff person.

Sample Display

Figure 88. shows the file system parameters display.

Figure 88. `samu`**(1M)** `N` **Display**

```
File system parameters                    samu 4.2         08:55:19 Sept 8
2004
License: License never expires.
mount_point     : /sam1                    partial         : 16k
fs_type         : 6                        maxpartial      : 16k
server          :                          partial_stage   : 16384
filesystem name: samfs1                    flush_behind    : 0
eq  type        : 10 ms                    stage_flush_beh: 0
state version   :    0    2                stage_n_window  : 262144
(fs,mm)_count   :    5    0                stage_retries   : 3
sync_meta       : 0                        stage timeout   : 0
stripe          : 0                        dio_consec r,w  :    0    0
mm_stripe       : 1                        dio_frm_min r,w:  256  256
high  low       :  90%  70%                dio_ill_min r,w:    0    0
readahead       : 1048576                  ext_bsize       : 4096
writebehind     : 524288
wr_throttle     : 16777216
rd_ino_buf_size: 16384
wr_ino_buf_size: 512
config          : 0x08520530               mflag           : 0x00000044
status          : 0x00000001

Device configuration:
ty    eq state   device_name                          fs family_set
md    11 on      /dev/dsk/c5t8d0s3                    10 samfs1
md    12 on      /dev/dsk/c5t8d0s4                    10 samfs1
```

Figure 88. `samu`**(1M)** `N` **Display (Continued)**

```
md    13 on      /dev/dsk/c5t8d0s5                     10 samfs1
md    14 on      /dev/dsk/c5t8d0s6                     10 samfs1
md    15 on      /dev/dsk/c5t8d0s7                     10 samfs1
```

(o) - Optical Disk Status Display

The `o` display shows the status of all optical disk drives configured within the StorageTek ASM environment.

To invoke this display, type the following command:

```
Command:o
```

Navigation

Table 36. shows the control keys you can use in this display.

Table 36. Control Keys for the `o` **Display**

Key	Function
Ctrl-b	Page backward
Ctrl-f	Page forward
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-k	Select (manual, automated library, both, priority)
Ctrl-u	Half-page backward

Sample Display

Figure 89. shows the optical disk status display.

Figure 89. `samu`**(1M)** `o` **Display**

```
Optical disk status           samu    4.2   Thu Oct 11 13:15:40

ty  eq  status      act  use  state  vsn
mo  35  --l---wo-r    1  29%  ready  oper2
```

Field Descriptions

Table 37. shows the field descriptions for this display.

Table 37. `samu`**(1M)** `o` **Display Field Descriptions**

Field	Description
`ty`	Device type.
`eq`	Equipment Ordinal of the optical disk.
`status`	Device status. For a description of status codes, see “Operator Display Status Codes” on page 175.
`act`	Activity count.
`use`	Percentage of cartridge space used.
`state`	Current operating state of the optical disk. Valid device states are as follows: • `ready`—The device is on, and the disk is loaded in the transport; available for access. • `notrdy`—The device is on, but no disk is present in the transport. • `idle`—The device is not available for new connections. Operations in progress continue until completion. • `off`—The device is not available for access. • `down`—The device is available only for maintenance access.
`vsn`	Volume serial name assigned to the optical disk, or the keyword `nolabel` if the volume is not labeled.

(p) - Removable Media Load Requests Display

The `p` display lists information about pending load requests for removable media. You can use the *mt* argument to select either a specific type of media, such as DLT tape, or a family of media, such as tape. The priority display lists the priority in the preview queue, rather than the user queue, and sorts the entries by priority.

It displays mount requests in the following formats:

- Both manual and automated library requests by user
- Both manual and automated library requests by priority
- Manual requests only

- Automated library requests only

You can invoke this display differently, depending on what you need to view, as follows:

- To display mount requests for all removable devices currently selected, type the command with the following format:

```
Command:p
```

- To display mount requests for devices of a given removable media type, type the command with the following format:

```
Command:p mt
```

For *mt*, specify one of the media types shown in the `mcf`(4) man page.

Navigation

Table 38. shows the control keys you can use in this display.

Table 38. Control Keys for the `p` Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-k	Toggle between the different display formats
Ctrl-u	Half-page backward

Sample Display

Figure 90. shows the removable media load requests display.

Figure 90. `samu`(1M) `p` Display

```
Removable media load requests all both  samu 4.2 09:14:19 Sept 8 2004
License: License never expires.                      count: 1

index type pid     user          rb   flags     wait count   vsn
   0  dt   15533   root          150  W--f---   0:00         DAT001
```

Field Descriptions

Table 39. shows the field descriptions for this display.

Table 39. `samu`(1M) `p` Display Field Descriptions

Field	Description
`index`	Index number in the preview table.
`type`	Device type code assigned to the removable media.
`pid`	UNIX process identifier. A process identifier of 1 indicates NFS access.
`user`	Name assigned to the user requesting the load.
`priority`	Priority of the request.
`rb`	Equipment Ordinal of the automated library in which the requested VSN resides.
`flags`	Flags for the device. See Table 40..
`wait`	The elapsed time since the mount request was received.
`count`	The number of requests for this VSN, if it is a stage.
`vsn`	Volume serial name of the volume.

Flags

Table 40. shows the flags for the `p` display.

Table 40. Flags Field for the `samu`(1M) `p` Display

Field	Description
`W------`	Write access requested
`-b-----`	Entry is busy
`--C----`	Clear VSN requested
`---f---`	File system requested
----N--	Media is foreign to the StorageTek ASM file system
`-----S-`	Flip side already mounted
`------s`	Stage request flag

(P) - Active Services Display

The `P` display lists the services registered with the StorageTek QFS and StorageTek ASM single port multiplexer.

To invoke this display, type the following command:

```
Command:P
```

Navigation

Table 41. shows the control keys you can use in this display.

Table 41. Control Keys for the P Display

Key	Function
Ctrl-b	Page backward
Ctrl-f	Page forward

This display is designed for debugging. It is intended to be used only with the assistance of a StorageTek support staff person.

Sample Display

Figure 91. shows the active services display.

Figure 91. samu**(1M)** P **Display**

```
Active Services                               samu     4.2    09:08:33 Sept 8 2004
License: License never expires.

Registered services for host 'pup':
    sharedfs.qfs2
    sharedfs.qfs1
  2 service(s) registered.
```

(r) - Removable Media Status Display

The r display enables you to monitor the activity on removable media devices such as tape drives. You can monitor either a specific type of device, such as video tape, or a family of devices such as all tape devices.

You can invoke this display differently, depending on what you need to view, as follows:

- To display the status for all removable media devices, type the command with the following format:

```
Command:r
```

- To display the status for a specific device, type the command with the following format:

```
Command:r eq
```

For *eq*, specify the Equipment Ordinal for the device.

Sample Display

Figure 92. shows the removable media status display.

Figure 92. `samu`**(1M)** `r` **Display**

```
Removable media status: all              samu 4.2         09:11:27 Sept 8
2004
License: License never expires.

ty   eq  status       act  use  state     vsn
dt  150  --l------r     0  63%  ready     DAT001
```

Field Descriptions

Table 42. shows the field descriptions for this display.

Table 42. `samu`**(1M)** `r` **Display Field Descriptions**

Field	Description
`ty`	Device type.
`eq`	Equipment Ordinal of the drive.
`status`	Device status. For a description of status codes, see “Operator Display Status Codes” on page 175.
`act`	Activity count.

Table 42. `samu`**(1M)** `r` **Display Field Descriptions (Continued)**

Field	Description
`use`	Percentage of cartridge space used.
`state`	Current operating state of the removable media. Valid device states are as follows: • `ready`—The device is on, and the disk or tape is loaded in the transport; available for access. • `notrdy`—The device is on, but no disk or tape is present in the transport. • `idle`—The device is not available for new connections. Operations in progress continue until completion. • `off`—The device is not available for access. • `down`—The device is available only for maintenance access.
`vsn`	Volume serial name assigned to the volume, or the keyword `nolabel` if the volume is not labeled. Blank if no volume is present in the transport, or device is off.

(`R`) - StorageTek ASM-Remote Information Display

The `R` display shows information and status on StorageTek ASM-Remote configurations.

To invoke this display, type the following command:

```
Command:R
```

This display is designed for debugging. It is intended to be used only with the assistance of a StorageTek support staff person.

(s) - Device Status Display

The `s` display shows the status for all devices configured within the StorageTek ASM environment.

To invoke this display, type the following command:

```
Command:s
```

Navigation

Table 43. shows the control keys you can use in this display.

Table 43. Control Keys for the `s` Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-u	Half-page backward

Sample Display

Figure 93. shows the device status display.

Figure 93. `samu`(1M) `s` Display

```
Device status                              samu     4.2    09:14:05 Sept 8 2004
License: License never expires.

ty      eq state   device_name                     fs status   pos
sk     100 on      /etc/opt/SUNWsamfs/dcstkconf   100 m--------r
         stk_dismount(2275) 0, volser 700073
sg     120 on      /dev/rmt/2cbn                  100 ---------p
         empty
sg     130 on      /dev/rmt/5cbn                  100 --l----o-r
         Ready for data transfer
sg     140 on      /dev/rmt/6cbn                  100 ---------p
         empty
sg     150 on      /dev/rmt/4cbn                  100 ---------p
         empty
hy     151 on      historian                      151 ----------
```

Field Descriptions

Table 44. shows the field descriptions for this display.

Table 44. `samu`(1M) `s` Display Field Descriptions

Field	Description
`ty`	Device type.
`eq`	Equipment ordinal of the device.
`state`	Current operating state of the device.

Table 44. `samu`**(1M)** `s` **Display Field Descriptions (Continued)**

Field	Description
`device_name`	Path to the device. For file system devices, this is the file system name.
`fs`	Equipment Ordinal of the family, set to which the device belongs.
`status`	Device status. For a description of status codes, see "Operator Display Status Codes" on page 175.

(S) - Sector Data Display

The `S` display shows raw device data.

To invoke this display, type the following command:

```
Command:S
```

Navigation

Table 45. shows the control keys you can use in this display.

Table 45. Control Keys for the S Display

Key	Function
Ctrl-b	Previous sector
Ctrl-d	Page forward (top portion)
Ctrl-f	Next sector
Ctrl-k	Advance display format
Ctrl-u	Page backward (top portion)

This display is designed for debugging. It is intended to be used only with the assistance of a StorageTek support staff person.

(t) - Tape Drive Status Display

The `t` display shows the status of all tape drives configured within the StorageTek ASM environment.

To invoke this display, type the following command:

```
Command:t
```

Navigation

Table 46. shows the control keys you can use in this display.

Table 46. Control Keys for the `t` Display

Key	Function
Ctrl-b	Page backward
Ctrl-f	Page forward

Sample Display

Figure 94. shows the tape drive status display.

Figure 94. `samu`(1M) `t` Display

```
Tape drive status                         samu     4.2    09:21:07 Sept 8 2004
License: License never expires.

ty   eq  status       act  use  state    vsn
sg  120  ---------p     0   0%  notrdy
        empty
sg  130  ---------p     0   0%  notrdy
        empty
sg  140  ---------p     0   0%  notrdy
        empty
sg  150  --l------r     0  41%  ready    700088
        idle
```

Field Descriptions

Table 47. shows the field descriptions for this display.

Table 47. `samu`(1M) `t` Display Field Descriptions

Field	Description
`ty`	Device type.
`eq`	Equipment Ordinal of the drive.
`status`	Device status. For a description of status codes, see “Operator Display Status Codes” on page 175.
`act`	Activity count.

Table 47. `samu`**(1M)** `t` **Display Field Descriptions (Continued)**

Field	Description
`use`	Percentage of cartridge space used.
`state`	Current operating state of the removable media. Valid device states are as follows: • `ready`—The device is on and the disk or tape is loaded in the transport; available for access. • `notrdy`—The device is on but no disk or tape is present in the transport. • `idle`—The device is not available for new connections. Operations in progress continue until completion. • `off`—The device is not available for access. • `down`—The device is available only for maintenance access.
`vsn`	Volume serial name assigned to the volume, or the keyword `nolabel` if volume is not labeled. Blank if no volume is present in the transport, or device is off.

(T) - SCSI Sense Data Display

The `T` display shows the SCSI status of a SCSI device.

To invoke this display, type the following command:

```
Command:T
```

Navigation

Table 48. shows the control keys you can use in this display.

Table 48. Control Keys for the `T` **Display**

Key	Function
Ctrl-b	Previous equipment
Ctrl-f	Next equipment

This display is designed for debugging. It is intended to be used only with the assistance of a StorageTek support staff person.

(`u`) - Staging Queue Display

The `u` display lists all files in the staging queue.

To invoke this display, type the following command:

```
Command:u
```

Navigation

Table 49. shows the control keys you can use in this display.

Table 49. Control Keys for the u Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-k	Display the path on the second line of each entry
Ctrl-u	Half-page backward

Sample Display

Figure 95. shows the staging queue display.

Figure 95. samu(1M) u Display

```
Staging queue by media type: all        samu 4.2          09:24:23 Sept 8
2004
License: License never expires.                          volumes 1 files 22

ty      length  fseq         ino   position      offset vsn

dt   451.611k    20        1030      207cc         473 DAT001
dt   341.676k    20        1031      207cc         7fc DAT001
dt   419.861k    20        1032      207cc         aa9 DAT001
dt   384.760k    20        1033      207cc         df2 DAT001
dt   263.475k    20        1034      207cc        10f5 DAT001
dt   452.901k    20        1035      207cc        1305 DAT001
dt   404.598k    20        1036      207cc        1690 DAT001
dt   292.454k    20        1037      207cc        19bb DAT001
dt   257.835k    20        1038      207cc        1c05 DAT001
dt   399.882k    20        1040      207cc        1e0b DAT001
dt   399.882k    40        1029      208d7           2 DAT001
dt   257.835k    40        1030      208d7         323 DAT001
dt   292.454k    40        1031      208d7         528 DAT001
dt   404.598k    40        1032      208d7         772 DAT001
dt   452.901k    40        1033      208d7         a9d DAT001
dt   263.475k    40        1034      208d7         e28 DAT001
dt   384.760k    40        1035      208d7        1038 DAT001
dt   419.861k    40        1036      208d7        133b DAT001
dt   341.676k    40        1037      208d7        1684 DAT001
```

Figure 95. `samu`**(1M)** `u` **Display (Continued)**

```
dt    451.611k     40         1038       208d7         1931  DAT001
dt    161.326k     40         1039       208d7         1cba  DAT001
dt    406.400k     40         1040       208d7         1dfe  DAT001
```

Field Descriptions

Table 50. shows the field descriptions for this display.

Table 50. `samu`**(1M)** `u` **Display Field Descriptions**

Field	Description
`ty`	Device type.
`length`	File length.
`fseq`	File system equipment number.
`ino`	The inode number.
`position`	The position of the archive file on the specific medium.
`offset`	Offset of the archive file on the specific medium.
`vsn`	Volume serial name of the volume.

(`U`) - Device Table Display

The `U` display shows the device table in a human-readable form.

You can invoke this display differently, depending on what you need to view, as follows:

- To display the device table for all devices, type the command with the following format:

```
Command:U
```

- To display the device table for a specific device, type the command with the following format:

```
Command:U eq
```

For *eq*, specify the Equipment Ordinal of the device.

Navigation

Table 51. shows the control keys you can use in this display.

Table 51. Control Keys for the U Display

Key	Function
Ctrl-b	Previous equipment
Ctrl-f	Next equipment

This display is designed for debugging. It is intended to be used only with the assistance of a StorageTek support staff person.

Sample Display

Figure 96. shows the device table display.

Figure 96. samu(1M) U Display

```
Device table: eq: 10    addr: 00000450  samu 4.2        09:28:40 Sept 8
2004
License: License never expires.

message:

0004000000014d58 0000000000000000       00000000 delay
0000000000000000 mutex                  00000000 unload_delay
00000aa8 next
73616d66 set:  samfs1
73310000
00000000
00000000
000a000a eq/fseq
08010801 type/equ_type
0000     state
00000000 st_rdev
00000000 ord/model
00000000 mode_sense
00000000 sense
00000000 space
00000000 capacity
00000000 active
00000000 open
00000000 sector_size
00000000 label_address
00000000 vsn:
00000000
00000000
00000000
```

Figure 96. samu(1M) U Display (Continued)

```
00000000 status: ----------
00000000 dt
73616d66 name: samfs1
```

(v) - Automated Library Catalog Display

The `v` display shows the location and VSN of all disks or tapes currently cataloged in the automated library.

You can invoke this display differently, depending on what you need to view, as follows:

- To display the catalog for all devices, type the command with the following format:

```
Command:v
```

- To display catalog information for a specific device, type the command with the following format:

```
Command:v eq
```

For *eq*, specify the Equipment Ordinal of the device. Type the keyword `historian` to view the historian catalog.

At certain times, `samu`(1M) prompts for a device to be entered, as follows:

```
Enter robot:eq
```

For *eq*, specify the Equipment Ordinal of the device or press return. Pressing return displays information for the previous device specified.

For a list of all device names and Equipment Ordinals, see "(c) - Device Configuration Display" on page 137.

Navigation

Table 52. shows the control keys you can use in this display.

Table 52. Control Keys for the v Display

Key	Function
Ctrl-b	Page backward.
Ctrl-d	Next library catalog.
Ctrl-f	Page forward.

Table 52. **Control Keys for the v Display (Continued)**

Key	Function
Ctrl-i	Detailed, 2-line display format. When you enter Ctrl-i once, it shows times and barcodes. When you enter Ctrl-i a second time, it shows volume reservations on the second line.
Ctrl-k	Advance sort key. After you enter Ctrl-k, you can enter one of the following to select a sort key: 1 - sort by slot. 2 - sort by count. 3 - sort by usage. 4 - sort by VSN. 5 - sort by access time. 6 - sort by barcode. 7 - sort by label time.
Ctrl-u	Previous automated library catalog.
/	Search for VSN
%	Search for barcode
$	Search for slot

Sample Display

Figure 97. shows the automated library catalog display.

Figure 97. `samu`**(1M)** `v` **Display**

```
Robot VSN catalog by slot       : eq 100samu     4.2    09:30:25 Sept 8
2004
License: License never expires.                                  count 32
slot          access time count use flags          ty vsn

   0     2004/05/08 08:35   64   0% -il-o-b-----   sg 700071
   1     2004/05/08 09:08   27  12% -il-o-b-----   sg 700073
   2     2004/05/08 09:12   26  12% -il-o-b-----   sg 700077
   3     2004/05/08 08:39   37  40% -il-o-b-----   sg 700079
   4     2004/05/08 09:16   24   6% -il-o-b-----   sg 700084
   5     2004/05/08 09:18   24  41% -il-o-b-----   sg 700088
   6     none                0   0% -il-o-b-----   sg 700090
   7     none                0   0% -il-o-b-----   sg 700092
   8     none                0   0% -il-o-b-----   sg 000155
   9     none                0   0% -il-o-b-----   sg 000156
  10     none                0   0% -il-o-b-----   sg 000157
  11     none                0   0% -il-o-b-----   sg 000158
```

Figure 97. `samu`**(1M)** `v` **Display (Continued)**

```
  12      none                     0    0% -il-o-b-----   sg 000154
  13      none                     0    0% -il-o-b-----   sg 000153
  14      none                     0    0% -il-o-b-----   sg 000152
```

Field Descriptions

Table 53. shows the field descriptions for this display.

Table 53. `samu`**(1M)** `v` **Display Field Descriptions**

Field	Description
`Robot VSN catalog`	Name of the specified automated library and time the display refreshed.
`count`	Number of slots allocated in this library's catalog.
`slot`	Slot number within the specified library.
`access time`	Time the volume was last accessed.
`count`	Number of accesses to this volume since the last audit operation.
`use`	Percentage of space used for the volume.
`flags`	Flags for the device. See Table 54. for information about the flags.
`ty`	Device type.
`vsn`	Volume serial name of the volume.

Flags

In some cases, more than one flag can occur in a field, and one flag overrides the other. Table 54. shows the flags from the `flags` field in Table 53..

Table 54. Flags Field for `samu`**(1M)** `v` **Display**

Flags	Description
`A-----------`	Volume needs audit.
`-i----------`	Slot in use.
`--l---------`	Labeled. Overrides `N`.
`--N---------`	Unlabeled. This volume is foreign to the StorageTek ASM environment.
`---E--------`	Media error. Set when the StorageTek ASM software detects a write error on a cartridge.
`----o-------`	Slot occupied.

Table 54. **Flags Field for** `samu`**(1M)** `v` **Display (Continued)**

Flags	Description
`-----C------`	Volume is a cleaning tape. Overrides `p`.
`-----p------`	Priority VSN.
`------b-----`	Barcode detected.
`-------W----`	Write protect. Set when the physical write protection mechanism is enabled on a cartridge.
`--------R---`	Read only.
`---------c--`	Recycle.
`----------d-`	Duplicate VSN. Overrides `U`.
`----------U-`	Volume unavailable.
`-----------f`	Archiver found volume full.
`-----------X`	Export slot.

(w) - Pending Stage Queue

The `w` display shows queued stage requests for which the volumes have not yet been loaded.

You can invoke this display differently, depending on what you need to view, as follows:

- To display the pending stage queue for all media, type the command with the following format:

```
Command:w
```

- To display the pending stage queue for a specific media type, type the command with the following format:

```
Command:w mt
```

For *mt*, specify one of the media types shown in the `mcf`(4) man page.

Navigation

Table 55. shows the control keys you can use in this display.

Table 55. Control Keys for the w Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-k	Display the path on the second line of each entry
Ctrl-u	Half-page backward

Sample Display

Figure 98. shows the pending stage queue.

Figure 98. `samu`**(1M)** `w` **Display**

```
Pending stage queue by media type: all      samu      4.2    Thu Oct 11
13:20:27
License: License never expires.                            volumes 1 files 13

ty      length  fseq  ino  position  offset  vsn

at      1.383M     1   42     3a786    271b  000002
at      1.479M     1   56     3a786    5139  000002
at   1018.406k     1   60     3a786    6550  000002
at      1.000M     1   65     3a786    7475  000002
at      1.528M     1   80     3a786    99be  000002
at      1.763M     1   92     3a786    ce57  000002
at      1.749M     1  123     3a786   11ece  000002
at    556.559k     1  157     3a786   1532f  000002
at    658.970k     1  186     3a786   17705  000002
at    863.380k     1  251     3a786   1dd58  000002
at      1.268M     1  281     3a786   1f2b7  000002
at      1.797M     1  324     3a786   23dfa  000002
at      1.144M     1  401     3a786   2bb6d  000002
```

Field Descriptions

Table 56. shows the field descriptions for this display.

Table 56. `samu`(1M) `w` Display Field Descriptions

Field	Description
`ty`	Device type.
`length`	File length.
`fseq`	File system Equipment Ordinal.
`ino`	The inode number.
`position`	The position (in decimal format) of the archive file on the specific medium.
`offset`	Offset of the archive file on the specific medium.
`vsn`	Volume serial name of the volume.

Operator Display Status Codes

The operator displays have different status codes for the removable media device displays and the file system displays. The following sections describe these displays.

Removable Media Device Display Status Codes

The `o`, `r`, `s`, and `t` operator displays show status codes for removable media devices. Status codes are displayed in a 10-position format, reading from left (position 1) to right (position 10).

The status codes in this section do not apply to the `samu`(1M) `f`, `m`, and `v` displays. For information about the status codes for the `f` and `m` displays, see "File System Display Status Codes" on page 176. For information about the status codes for the `v` display, see "(v) - Automated Library Catalog Display" on page 170.

Table 57. defines the valid status codes for each position.

Table 57. Removable Media Device Display Status Codes

Status Bit	Meaning for a Device
`s---------`	Media is being scanned.
m---------	The automated library is operational.
`M---------`	Maintenance mode.
`-E--------`	Device received an unrecoverable error in scanning.
`-a--------`	Device is in audit mode.

Table 57. Removable Media Device Display Status Codes (Continued)

Status Bit	Meaning for a Device
`--l-------`	Media has a label.
`--N-------`	Foreign media.
--L-------	Media is being labeled.
`---I------`	Waiting for device to idle.
`---A------`	Needs operator attention.
`----C-----`	Needs cleaning.
`----U-----`	Unload has been requested.
`-----R----`	Device is reserved.
`------w---`	A process is writing on the media.
`-------o--`	Device is open.
`--------P-`	Device is positioning (tape only).
`--------F-`	For automated libraries, all storage slots occupied. For tape and magneto-optical drives, media is full.
`---------R`	Device is ready and the media is read-only.
`---------r`	Device is spun up and ready.
`---------p`	Device is present.
`---------W`	Device is write protected.

File System Display Status Codes

The `f` and `m` operator displays show status codes for file systems. Status codes are displayed in an 11-position format, reading from left (position 1) to right (position 11).

The status codes in this section do not apply to the `samu`(1M) `c`, `o`, `r`, `s`, `t`, or `v` displays. For information about the status codes for the `c`, `o`, `r`, `s`, and `t` displays, see “Removable Media Device Display Status Codes” on page 175. For information about the status codes for the `v` display, see “(v) - Automated Library Catalog Display” on page 170.

Table 58. defines the valid status codes for each position.

Table 58. File System Display Status Codes

Status Bit	Meaning for a File System
`m----------`	File system is currently mounted.
`M----------`	File system is being mounted.
`-u---------`	File system is being unmounted.

Table 58. File System Display Status Codes (Continued)

Status Bit	Meaning for a File System
`--A--------`	File system data is being archived.
`---R-------`	File system data is being released.
`----S------`	File system data is being staged.
`-----1-----`	StorageTek ASM file system version 1.
`-----2-----`	StorageTek ASM file system version 2.
`------c----`	StorageTek QFS shared file system.
`-------W---`	Single writer.
`--------R--`	Multireader.
`---------r-`	`mr` devices.
`----------d`	`md` devices.

Operator Display Device States

The `c`, `m`, `o`, `r`, `s`, and `t` operator displays show device state codes. These codes represent the current access state for the device. Table 59. defines the valid state codes.

Table 59. Operator Display Device States

Device State	Description
`on`	The device is available for access. For certain displays, this state might be superseded by the states `ready` or `notrdy`.
`ro`	The device is available for read-only access. For certain displays, this state might be superseded by the states `ready` or `notrdy`.
`off`	The device is not available for access. For tape and optical disk drives, possible reasons for the device to be in the `off` state include the following: • Cleaning was requested, but no cleaning cartridge was found in the automated library. • The cleaning cartridge cannot be loaded or unloaded from the drive. • Initialization found the drive status to be full, and attempts to clear the drive failed. • The system was unable to clear a cartridge from a drive. • Opening the drive for I/O failed during spin-up. • An error other than `NOT READY` was received when spinning down the drive for unloading. • Opening the standard tape driver on the drive failed during spin-up.
`down`	The device is available for maintenance access only.
`idle`	The device is not available for new connections. Operations in progress continue until completion.
`ready`	The device is on and the disk or tape loaded in the transport is available for access.
`notrdy`	The device is on, but no disk or tape is present in the transport.
`unavail`	The device is unavailable for access and cannot be used for automatic StorageTek ASM operations. You can continue to use the load(1M) and unload(1M) commands for moving media while the device is in the `unavail` state.

You can use the `samu`(1M) `down`, `off`, and `on` device state commands to change device states to `down`, `off`, or `on`. You can enter these commands from any `samu`(1M) display, but if you enter them from the `c`, `m`, `o`, `r`, `s`, or `t` display, you can see the device state change in the display. For example, you could set a device state to `off` from within the `P` display, but you would not be able to see the new device state reflected in the display.

The following procedures show what to type to change a device's state from `down` to `on` and from `on` to `down`.

To Change a Drive State from down to on

1. Bring up a `samu`(1M) display that shows drive and automated library device states.

   The following `samu`(1M) displays all show device states: `c`, `m`, `o`, `r`, `s`, and `t`.

2. Visually inspect the display to verify that the device is in the `down` state.

3. Type `:off`.

   Turning the device off halts all activity so the device can be started cleanly in the next step. For example:

```
Command:off eq
```

   For *eq*, specify the Equipment Ordinal of the device.

4. Type `:on`.

   For example:

```
Command:on eq
```

   For *eq*, specify the Equipment Ordinal of the device.

To Change a Drive State from on to down

1. Bring up a `samu`(1M) display that shows drive and automated library device states.

   The following `samu`(1M) displays all show device states: `c`, `m`, `o`, `r`, `s`, and `t`.

2. Visually inspect the display to verify that the device is in the `on` state.

3. Type `:off`.

   Turning the device off halts all activity so the device can be stopped cleanly in the next step. For example:

```
Command:off eq
```

For *eq*, specify the Equipment Ordinal of the device.

4. Type `:down`.

   For example:

```
Command:down eq
```

   For *eq*, specify the Equipment Ordinal of the device.

Operator Commands

The following sections describe the operator commands that you can enter from the `samu`(1M) operator utility's command interface. You can enter the commands from any display.

The following types of operator commands are available:

- "Device Commands" on page 181
- "ASM Commands — Archiver Control" on page 182
- "ASM Commands — Stager Control" on page 185
- "ASM Commands — Releaser Control" on page 184
- "File System Commands — I/O Management" on page 186
- "File System Commands — Direct I/O Management" on page 188
- "File System Commands — StorageTek QFS Shared File Systems" on page 190
- "File System Commands - Miscellaneous" on page 191
- "Automated Library Commands" on page 193
- "Miscellaneous Commands" on page 195

If you want to enter any operator commands from the Solaris operating system (OS) command line, you must use them as arguments to the `samcmd`(1M) command. For more information about the `samcmd`(1M) command, see the `samcmd`(1M) man page.

In the following subsections, each `samu`(1M) command is prefaced with a colon (:) when it is entered to designate that a command is being entered and not a series of hot keys.

Device Commands

Table 60. shows the device commands and their actions.

Table 60. Device Command Actions

Command	Action
down	Terminates operation on device *eq*.
idle	Restricts access to device *eq* by preventing new connections to the device. Existing operations continue until completion.
off	Logically turns off device *eq*.
on	Logically turns on device *eq*.
unavail	Selects device *eq* and makes it unavailable for use with the StorageTek ASM file system. You might set a drive state to `unavail`, for example, in a disaster recovery situation in which you are trying to load media to restore a file system and you do not want the StorageTek ASM software to attempt to use this drive.
unload	Unloads the mounted media for the specified removable media device *eq*. For magazine devices, the `unload` command unloads the mounted cartridge and ejects the magazine.

Figure 99. shows the formats for the device control commands.

Figure 99. Formats for the Device Control Commands

```
:down eq
:idle eq
:off eq
:on eq
:unavail eq
:unload eq
```

For *eq*, specify the Equipment Ordinal of the device.

ASM Commands — Archiver Control

Table 61. shows the archiver commands and their actions.

Table 61. Archiver Command Actions

Command	Action
`aridle`	Stops all archiving at the next convenient point. For example, at the end of the current `tar`(1) file for `sam-arcopy` operations. You can also use this command to stop all archiving activity for all file systems prior to unmounting the file systems.
arrerun	Performs a soft restart on the archiver. The archiver daemons are restarted, and all work in progress is recovered.
`arrestart`	Interrupts the archiver and restarts the archiver. This action occurs regardless of the state of the archiver. Therefore, use `arrestart` with caution. Some copy operations to archive media might not complete and must be repeated. This wastes space on the media.
arrmarchreq	Removes an archive request.
arrun	Causes the archiver to begin archiving. This command overrides any existing global `wait` command in the `archiver.cmd` file.
arscan	Scans the file system.
`arstop`	Stops all archiving immediately.
artrace	Performs archiver tracing.

Figure 100. shows the formats for the archiver commands.

Figure 100. Formats for the Archiver Commands

```
:aridle [ dk | rm | fs.fsname ]
:arrerun
:arrestart
:arrmarchreq fsname.[* | archreq]
:arrun [ dk | rm | fs.fsname ]
:arscan fsname[.dir | ..inodes][int]
:arstop [ dk | rm | fs.fsname ]
:artrace [fs.fsname]
```

The arguments to these commands are optional. If no arguments are specified, all file systems are affected. If arguments are specified, the command takes action based on the type of archive file specified (`dk` or `rm`) or the file system specified. Table 62. shows the archiver command arguments.

Table 62. Archiver Command Arguments

Argument	Description
`dk`	Specifies that this command pertains to disk archive files.
`rm`	Specifies that this command pertains to removable media files.
fsname	Specifies that this command pertains to a specific file system. Enter a file system name for *fsname*.
archreq	Specifies the name of a specific archive request file in the following format: *arset*.*copy*.*seq_num* This file name has three components. Use a period to separate each component. The first component is the name of the archive set. The second component is the copy number (1, 2, 3, or 4). The third component is a sequence number that the archiver assigns. More than one archive request can exist at one time. You can use the `showqueue`(1M) command to obtain the names of the archive request files in the system. Figure 101. shows how to use this command. Alternatively, you can go to the following directory and list the files present: `/var/opt/SUNWsamfs/archiver/`***fsname***`/ArchReq`
`*`	Signifies all files.
dir	Specifies a specific directory name. This is the directory to scan.
`.inodes`	Specifies that the inodes should be scanned.
int	An integer number of seconds to delay the scan.

Figure 101. shows using the `showqueue`(1M) command to obtain an *archreq* file name that can be used as input to the `arrmarchreq samu`(1M) command.

Figure 101. Using `showqueue`(1M)

```
# showqueue samfs9
Filesystem samfs9:
Scan list: empty
Archive requests
arset1.2.0 schedule 2004-01-22 16:23:07
    files:697 space:   4.934G flags: offline
 (min:    1.000k)    priority: 0 0
```

Figure 101. Using `showqueue`(1M) (Continued)

```
    No volumes available
    Drive 1
      Files: 695, bytes:    1.932G (min:    1.000k)
  Stage volumes:
   lt.CFX600
   lt.CFX601

arset1.1.1 schedule 2004-01-22 16:23:07
    files:3 space:   6.236M flags:
 (min:  826.000k)    priority: 0 0
    No volumes available
    Drive 1
      Files: 3, bytes:    6.236M (min:  826.000k)
```

Figure 101. shows that files `arset1.2.0` and `arset1.1.1` are archive request files.

The :hwm_archive eq and :nohwm_archive eq Commands

The `hwm_archive` command invokes the archiver when the amount of data in the file system increases to a level above the releaser's high-water mark. You can set the releaser's high-water mark by using the `thresh` command. For information about the `thresh` command, see "The :thresh eq high low Command" on page 184. The `nohwm_archive` command disables this capability and is the default.

For *eq*, specify the Equipment Ordinal for the file system.

The :thresh eq high low Command

The `thresh` command sets the high and low thresholds for a file system to control file archiving.

For *eq*, specify the Equipment Ordinal of the storage family set.

For *high*, specify the high threshold.

For *low*, specify the low threshold.

For example, the following command sets the high threshold to 50 percent and the low threshold to 40 percent for the storage Family Set whose file system Equipment Ordinal is 10:

```
:thresh 10 50 40
```

ASM Commands — Releaser Control

The following commands allow you to control aspects of the partial release feature. For more information about the partial release feature, see the

Releasing chapter in the *StorageTek ASM Storage and Archive Manager Guide*.

The :maxpartial eq value Command

The `maxpartial` command sets the maximum partial release size for the file system to *value* kilobytes. The partial release size cannot be set larger than this `maxpartial` setting.

For *eq*, specify the Equipment Ordinal for the file system.

For *value*, specify an integer such that `0` ≤ *value* ≤ `2097152`.

The :partial eq value Command

The `partial` command sets the number of kilobytes to leave online after release of the file. For more information, see the Releasing chapter in the *StorageTek ASM Storage and Archive Manager Guide*.

For *eq*, specify the Equipment Ordinal for the file system.

For *value*, specify the number of kilobytes to leave online. The default size is 16.

ASM Commands — Stager Control

The following commands allow you to control staging activities.

The :partial_stage eq value Command

The `partial_stage` command sets the partial stage size for the file system to *value* kilobytes. For a file with the partial release attribute, *value* specifies the file offset past which access results in the entire file being staged to disk.

For *eq*, specify the Equipment Ordinal for the file system.

For *value*, specify an integer that is greater than `0` but is less than the *value* specified for the `maxpartial` setting. For more information about the `maxpartial` setting, see "The :maxpartial eq value Command" on page 185. For more information about the partial release feature, see the Releasing chapter in the *StorageTek ASM Storage and Archive Manager Guide*.

The :stage_flush_behind eq value Command

The `stage_flush_behind` command sets the maximum stage flush-behind value. Pages being staged are written to disk asychronously to help the Solaris VM layer keep the pages clean.

For *eq*, specify the Equipment Ordinal for the file system.

For *value*, specify an integer number of kilobytes such that `0` ≤ *value* ≤ `8192`. By default, *value*=`0`, which disables `stage_flush_behind`.

The :stage_n_window eq value Command

The `stage_n_window` command works with the `stage`(1) command's `-n` option. This `samu`(1M) command sets the `stage`(1) command's `-n` option for the file system to *value*. This command is effective for files read directly from the archive media and for which `stage -n` has been specified. For a file with the `stage -n` attribute set, *value* is the amount of data that is staged to the application's buffer at any one time.

For *eq*, specify the Equipment Ordinal for the file system.

For *value*, specify an integer such that `64` ≤ *value* ≤ `2097152` kilobytes. The default is `256` for all for all file systems except for the StorageTek QFS shared file system, which is set to the value of the `minallocsz` mount option.

The :stage_retries eq value Command

The `stage_retries` command sets the number of stage retries attempted per archive copy when certain errors are encountered.

For *eq*, specify the Equipment Ordinal for the file system.

For *value*, specify a number such that `0` ≤ *value* ≤ `20`. When *value*=`0`, retries are not attempted. The default is `3`.

The :stclear mt.vsn Command

The `stclear` command clears a stage request.

For *mt*, specify the media type; for information about valid media types, see the `mcf`(4) man page.

For *vsn*, specify the volume to mount.

The :stidle Command

The `stidle` command idles the stager. Use this command if you want the stager to finish its current tasks and not commence any additional staging.

The :strun Command

The `strun` command restarts staging activity. You can use this command to restart the stager after you have issued the `stidle` command.

File System Commands — I/O Management

The following commands allow you to manage I/O characteristics dynamically.

The :flush_behind eq value Command

The `flush_behind` command sets the maximum `flush_behind` *value*. When set to a value greater than 0, modified pages that are being written sequentially are written to disk asynchronously to help the Solaris kernel layer keep the pages clean. This option sets the maximum `flush_behind` value.

For *value*, specify an integer number of kilobytes such that `0` ≤ *value* ≤ `8192`. By default, *value*=`0`, which disables `flush_behind`.

For *eq*, specify the Equipment Ordinal for the file system.

The :force_nfs_async eq and :noforce_nfs_async eq Commands

These commands allow you to control whether the file system caches NFS data written to the server even if NFS has requested that the data be written synchronously through to disk. The `force_nfs_async` command caches NFS data. The `noforce_nfs_async` command, which is the default, synchronously writes data through to disk.

The `force_nfs_async` command is effective only if the file system is mounted as an NFS server and only if the clients are mounted with the `noac` NFS mount option. For more information about mounting an NFS file system, see the `mount_nfs`(1M) man page.

For *eq*, specify the Equipment Ordinal for the file system.

CAUTION: The `force_nfs_async` option violates NFS protocols. Use this command with caution. In the event of a server interruption, data can be lost. Data is cached on the NFS server and cannot be seen immediately by all the clients if there are multiple NFS servers. Multiple NFS servers can be enabled within the StorageTek QFS shared file system. For more information about the StorageTek QFS shared file system, see "StorageTek QFS Shared File System" on page 85.

The :readahead eq contig Command

The `readahead` command specifies the maximum number of bytes that can be read ahead by the file system.

For *eq*, specify the Equipment Ordinal for the file system.

For *contig*, specify units of 1-kilobyte blocks. This must be an integer such that 1 < *contig* < 8192. The *contig* specified is truncated to a multiple of 8 kilobytes. The default *contig* is 8 (131072 bytes).

For example, the following command sets the maximum contiguous block size to 262,144 bytes for the file system defined as Equipment Ordinal `3`:

```
:readahead 3 256
```

This value can also be configured in the `samfs.cmd` file by specifying the readahead directive. For more information, see the `samfs.cmd`(4) man page.

The :sw_raid eq and :nosw_raid eq Commands

These options specify whether or not the file system aligns the writebehind buffer. Specify `sw_raid` if the software RAID feature of a package such as Solstice DiskSuite is also used on this file system. The default setting is `nosw_raid`.

For *eq*, specify the Equipment Ordinal for a file system.

The :writebehind eq contig Command

The `writebehind` command specifies the maximum number of bytes that can be written behind by a file system.

For *eq*, specify the Equipment Ordinal for a file system.

For *contig*, specify units of 1-kilobyte blocks. This must be an integer such that 1 < *contig* < 8192. The default *contig* is 8 (131072 bytes).

For example, the following command sets the maximum contiguous block size to 262,144 bytes for the file system defined as Equipment Ordinal `50`:

```
:writebehind 50 256
```

This value can also be configured in the `samfs.cmd` file by specifying the `writebehind` directive. For more information, see the `samfs.cmd`(4) man page.

The :wr_throttle eq value Command

The `wr_throttle` command sets the number of outstanding write bytes for one file to *value* kilobytes.

For *eq*, specify the Equipment Ordinal for a file system.

For *value*, specify an integer number of kilobytes. If *value*=`0`, there is no limit. The default is `16384`.

File System Commands — Direct I/O Management

The commands in this section control I/O on StorageTek QFS and StorageTek ASM file systems. They allow you to change the type of I/O for an individual file based on I/O size and history. If direct I/O is specified for a file, for example, through the `setfa`(1) command, these options are ignored and all I/O to regular files is direct, if possible.

These commands refer to both well-aligned and misaligned I/O. *Well-aligned* I/O occurs when the file offset falls on a 512-byte boundary and when the length of the I/O transfer is at least 512 bytes. *Misaligned* I/O occurs when the file offset does not fall on a 512-byte boundary and the length of the transfer is less than 512 bytes.

For more information about I/O and I/O management, see "Advanced Topics" on page 281.

The :dio_rd_form_min eq value and :dio_wr_form_min eq value Commands

These commands set the lower limits for well-aligned I/O to *value* 1024-byte blocks. Use the `dio_rd_form_min` command to set the *value* for reads, and use the `dio_wr_form_min` command to set the *value* for writes.

For *eq*, specify the Equipment Ordinal for the file system.

For *value*, specify an integer number of 1024-byte blocks to use for the lower limit. By default, *value*=`256`. If *value*=`0`, automatic I/O switching is disabled.

The :dio_rd_ill_min eq value and :dio_wr_ill_min eq value Commands

These commands set the lower limit for misaligned I/O to *value* 1024-byte blocks. Use the `dio_rd_ill_min` command to set the *value* for reads, and use the `dio_wr_ill_min` command to set the *value* for writes.

For *eq*, specify the Equipment Ordinal for the file system.

For *value*, specify an integer number of 1024-byte blocks to use for the lower limit. By default, *value*=`256`. If *value*=`0`, automatic I/O switching is disabled.

The :dio_rd_consec eq value and :dio_wr_consec eq value Commands

These commands set the number of consecutive I/O transfers that can occur, with a buffer size greater than the specified lower limits, to *value* operations.

For *eq*, specify the Equipment Ordinal for the file system.

For *value*, specify the the number of consecutive I/O transfers with a buffer size greater than the specified lower limit. The specified lower limit is the *value* of `dio_rd_form_min` for aligned reads or `dio_rd_ill_min` for misaligned reads. By default, *value*=0, which means that no default direct reads occur based on I/O sizes.

For more information, see one or more of the following commands or mount parameters:

- "The :dio_rd_form_min eq value and :dio_wr_form_min eq value Commands" on page 189
- "The :dio_rd_ill_min eq value and :dio_wr_ill_min eq value Commands" on page 189

The :forcedirectio eq and :noforcedirectio eq Commands

These commands allow you to control whether direct I/O be used as the default I/O mode. By default, the I/O mode is buffered and uses the page cache. The `forcedirectio` command enables direct I/O for all transfers. The `noforcedirectio` command enables the default, which is buffered I/O.

For *eq*, specify the Equipment Ordinal for the file system.

When direct I/O is specified, the system transfers data directly between the user's buffer and disk. Use direct I/O only if the file system is used for large, block-aligned, sequential I/O.

For more information about I/O, see "Advanced Topics" on page 281.

File System Commands — StorageTek QFS Shared File Systems

The following file system commands are supported on StorageTek QFS shared file systems only.

The :meta_timeo eq interval Command

The `metatimeo` command sets the StorageTek QFS shared file system metadata cache time out value. For more information about using this feature, see "Retaining Cached Attributes: the meta_timeo=n Option" on page 112.

For *eq*, specify the Equipment Ordinal of the file system.

For *interval*, specify an interval in seconds. The default *interval* is 15. After this interval expires, the client host systems obtain a new copy of the metadata information from the metadata server host.

The :mhwrite eq and :nomh_write eq Commands

These commands enable or disable multihost reads and writes. For information about this feature, see "Enabling Multiple Host Reads and Writes: the mh_write Option" on page 110.

For *eq*, specify the Equipment Ordinal of the file system.

The :minallocsz eq value and :maxallocsz eq value Commands

These commands set the minimum and maximum block allocation size.

For *eq*, specify the Equipment Ordinal of the file system.

For *value*, and for more information about this feature, see "Tuning Allocation Sizes: the minallocsz=n and maxallocsz=n Options" on page 109.

The :rdlease eq interval, :wrlease eq interval, and :aplease eq interval Commands

These commands tune the amount of time granted for read, write, and append leases. For information about this feature, see "Using Leases in a StorageTek QFS Shared File System: the rdlease=n, wrlease=n, and aplease=n Options" on page 109.

For *eq*, specify the Equipment Ordinal of the file system.

For *interval*, specify an integer number of seconds. All three leases enable you to specify an *interval* such that `15` ≤ *interval* ≤ `600`. The default *interval* is `30`.

File System Commands - Miscellaneous

The following commands allow you to control leases, allocation sizes, and various other file system characteristics.

The :invalid eq interval Command

The `invalid` command specifies that the file system hold cached attributes for at least *interval* seconds after a file is modified. You can specify this command only if the file system was mounted originally with the `reader` mount option. For information about mount options, see the `mount_samfs`(1M) man page.

For *eq*, specify the Equipment Ordinal for the file system.

For *interval*, specify the number of seconds to hold the attributes after file modification. For example, assume that *interval*=`30`. In such a file system, if you issue an `ls`(1) command, you might not see a newly created file appear in its output for 30 seconds after it has been created on its writer host.

The :mm_stripe eq value Command (StorageTek QFS File Systems Only)

The `mm_stripe` command sets the metadata stripe width for the file system to *value* 16-kilobyte disk allocation units (DAUs).

For *eq*, specify the Equipment Ordinal of the file system.

For *value*, specify either `0` or `1`. If *value*=`1`, which is the default, the file system writes one DAU of metadata to one LUN before switching to another LUN. If *value*=`0`, the metadata is round-robined across all available metadata LUNs.

The :qwrite eq and :noqwrite eq Commands (StorageTek QFS File Systems Only)

The `qwrite` and `noqwrite` commands control the ability to perform simultaneous reads and writes to the same file from different threads. Specify qwrite only if file system users handle multiple simultaneous transactions to the same file. For example, this is useful in database applications. The qwrite feature improves I/O performance by queuing multiple requests at the drive level. The qwrite specification is disabled for NFS reads or writes of the file system.

The default setting is noqwrite, so the file system disables simultaneous reads and writes to the same file. This is the mode defined by the UNIX vnode interface standard that gives exclusive access to only one writer and forces other writers and readers to wait.

For *eq*, specify the Equipment Ordinal of the file system.

The :refresh_at_eof eq and :norefresh_at_eof eq Commands (StorageTek QFS File Systems Only)

The refresh_at_eof and norefresh_at_eof commands can be used for fast updates to a StorageTek QFS multireader file system on hosts that are mounted with the reader mount option in a multireader file system. This option ensures that the system refreshes the current file size when the read buffer exceeds the end of file. You can use this, for example, if the writer host system is appending to a file and the reader is issuing tail(1) commands with the -f option. The default is norefresh_at_eof.

For *eq*, specify the Equipment Ordinal of the file system.

The :setuid eq and :nosetuid eq Commands

The setuid and nosetuid commands control whether setuid execution is allowed for this file system. These mount options control whether running programs are allowed to automatically change their owner IDs. For more information about the implications of using these mount options, see the suid and nosuid mount option descriptions on the mount_ufs(1M) man page and see the setuid(2) man page.

For *eq*, specify the Equipment Ordinal of the file system.

The :stripe eq value Command

The stripe command sets the stripe width for the file system to *value* disk allocation units (DAUs). The stripe width specifies that *value* multiplied by the DAU bytes are written to one LUN before switching to the next LUN. You can use the sammkfs(1M) -a command to set the DAU size on the file system when it is initialized.

For *eq*, specify the Equipment Ordinal of the file system.

For *value*, specify an integer such that 0 < *value* < 255. If *value*=0, files are round-robin on each slice. The default *value* on file systems with an ms Equipment Type and on file systems with an ma Equipment Type with no striped group (g*XXX*) components is as follows:

- 128 kilobytes/DAU for DAUs < 128 kilobytes
- 1 for DAUs > 128 kilobytes

By default, *value*=0 on a StorageTek QFS shared file system.

By default, *value*=0 on file systems with an ma Equipment Type with any striped group (g*XXX*) components.

The system sets *value*=0 if mismatched striped groups exist.

For more information about file system types, see "File System Design" on page 7 and see "Volume Management" on page 35.

The :sync_meta eq value Command

The sync_meta command determine whether metadata is written to disk every time it changes. If you are using this command on StorageTek QFS shared file system, also see "Specifying the Frequency With Which Metadata is Written: the sync_meta=n Option" on page 113.

For *eq*, specify the Equipment Ordinal of the file system.

For *value*, specify either 0 or 1, as follows:

- If *value* is 0, metadata is held in a buffer after it changes. For an unshared StorageTek QFS or StorageTek ASM file system in which higher performance is desired, you can set *value* to 0. In this case, the system performs a delayed write in which metadata is held in a buffer before it is written to disk. This is the default for unshared file systems and for file systems that are not mounted as multireader file systems.
- If *value* is 1, metadata is written to disk every time it changes. This slows performance, but it increases data consistency. This is the default for StorageTek QFS file systems mounted as multireader file systems or as shared file systems. For a StorageTek QFS shared file system, *value* must be set to 1 if failover capability is required.

The :trace eq and :notrace eq Commands

The trace command enables tracing for a file system. The notrace command disables tracing. These are global directives that affect all operations. For more information about file system tracing, see the defaults.conf(4) man page.

For *eq*, specify the Equipment Ordinal of a file system.

Automated Library Commands

The following commands control media activities in an automated library.

The :audit [-e] eq [:slot [:side]] Commands

The `audit` command causes the specified automated library device to mount each volume, read the VSN, and rebuild the library catalog.

If `-e` is specified, and the volume is on a tape cartridge, the tape skips to the end of data (EOD) and updates the space available. Note that the skip to EOD is not interruptible. Under certain conditions, it can take hours to complete.

For *eq*, specify the Equipment Ordinal of an automated library device.

For *slot*, specify the slot number containing the volume you want to load.

For *side*, specify the side of a magneto-optical disk. Must be `1` or `2`. This argument is not applicable to tape cartridges.

This command is not supported for network-attached libraries.

The :export eq:slot and :export mt.vsn Commands

The `export` command causes the specified automated library to export a volume to the mail slot. The volume is identified by its slot position within the automated library.

- If exporting by Equipment Ordinal and slot number, the specified automated library moves the volume to the mail slot. For *eq*, specify the Equipment Ordinal or device name. For *slot*, specify the slot number containing the volume you want to export.
- If exporting by logical identifier, the specified automated library moves the volume to the mail slot. For *mt*, specify the media type; for information about valid media types, see the `mcf`(4) man page. For *vsn*, specify the volume to export.

The :import eq Command

The `import` command causes the specified automated library to allow you to add a cartridge. For *eq*, specify the Equipment Ordinal of the automated library.

The :load eq:slot [:side] and :load mt.vsn Commands

The `load` command enables you to load by either a physical or a logical identifier, as follows:

- If loading by Equipment Ordinal and slot number, the specified automated library loads the volume into a drive.

  For *eq*, specify the Equipment Ordinal or device name.

  For *slot*, specify the slot number containing the volume you want to load.

For *side*, specify the side of a magneto-optical disk. Must be `1` or `2`. This argument is not applicable to tape cartridges.

- If loading by logical identifier, the specified automated library to load mounts a labeled volume into a drive.

  For *mt*, specify the media type; for information about valid media types, see the `mcf`(4) man page.

  For *vsn*, specify the volume to mount.

The :priority pid newpri Command

The `priority` command sets the load priority for a process. You can specify this command from the removable media mount requests display. For more information, see "(p) - Removable Media Load Requests Display" on page 157.

For *pid*, specify the priority shown in the `p` display.

For *newpri*, specify the priority you want to give the request. This should be an integer number.

Miscellaneous Commands

The following commands allow you to control tracing, open access to a disk device, and perform several other miscellaneous tasks.

The :clear vsn [index] Command

The `clear` command clears the specified VSN from the removable media mount requests display. For more information, see "(p) - Removable Media Load Requests Display" on page 157.

For *vsn*, specify the volume to mount. Any process waiting for the VSN mount is aborted.

For *index*, specify the decimal ordinal of the VSN in the removable media display.

The :devlog eq [option] Command

The `devlog` command sets one or more events to be logged.

For *eq*, specify the Equipment Ordinal of a device.

For *option*, specify one or more event types. Possible event types are as follows: `all`, `date`, `default`, `detail`, `err`, `event`, `label`, `mig`, `module`, `msg`, `none`, `retry`, `stage`, `syserr`, and `time`. For information about these options, see the `defaults.conf`(4) man page. If no *option* is specified, the system does not change the current events being logged for the *eq* specified.

The :dtrace Commands

The `dtrace` commands are as follows:

- `:dtrace` *daemon_name* `on`
- `:dtrace` *daemon_name* `off`
- `:dtrace` *daemon_name*`.`*variable value*

The `dtrace` commands specify various tracing options. Table 63. shows the tracing control command arguments.

Table 63. Tracing Command Arguments

Argument	Description
daemon_name	Specify the keyword `all` or a process name. If the keyword `all` is specified, the tracing command affects all daemons. If one of the following process names is specified, the tracing command affects that process only: `sam-archiverd`, `sam-catserverd`, `sam-fsd`, `sam-rftd`, `sam-recycler`, `sam-sharefsd`, and `sam-stagerd`. One of the keywords `on` or `off` can be specified after a process name. If `on` or `off` is specified, tracing is turned off or on for all processes specified.
variable value	Many different *variable* and *value* arguments can be specified. The `defaults.conf`(4) man page contains comprehensive information about these arguments. Specify one of the following *variable* and *value* combinations: • `file` *value*. For *value*, specify the name of a file to which trace files can be written. This can be a full path name. • `options` *value*. For *value*, specify a space-separated list of trace options. • `age` *value*. For *age*, specify the trace file rotation age. • `size` *value*. For *value*, specify the size of the trace file at which rotation will begin.

The :fs fsname Command

The `fs` command sets the file system to be displayed through the `N` display.

For `fsname`, specify the name of the file system to be examined.

The :mount mntpt Command

The `mount` command selects a StorageTek QFS or StorageTek ASM file system. For *mntpt*, specify the mount point of a file system.

The :open eq Command

The `open` command enables access to the specified disk device. You must issue this command before you can use the `read` command, disk sector display (`S`), or file label display (`F`).

For *eq*, specify the Equipment Ordinal of a device.

The :read addr Command

The `read` command reads the specified sector from the currently opened disk device. You must open the device before it can be read.

For *addr*, specify the hexadecimal sector address.

The :refresh i Command

The `refresh` command determines the amount of time between `samu`(1M) screen refreshes.

For *i*, specify a time in seconds.

The :snap [filename] Command

The `snap` command sends a snapshot of a display window to *filename*, which is the name of a file to receive the display information.

To aid in problem reporting, you can take a snapshot of all the `samu`(1M) utility's displays. Each new snapshot is appended to the snapshots file. The default file is `snapshots` in the current working directory. The file can be printed, examined using `vi`(1), or faxed to StorageTek customer support staff.

The :! shell_command Command

The `!` command enables you to run a shell command without leaving the `samu`(1M) operator utility.

File System Quotas

7

File system quotas control the amounts of online and total disk space that can be consumed by a specific user, a group of users, or an admin set in a file system. An *admin set* is a site-determined group of users.

Quotas help control the size of a file system by limiting the amount of space and the number of inodes each user can consume. Quotas can be especially useful on file systems that contain user home directories. After quotas are enabled, you can monitor usage and adjust the quotas as needs change.

This chapter contains the following sections:

Overview

You can set file system quotas on a user, group, or a site-defined admin set basis. You, the system administrator, can set limits on the number of files, the number of blocks online, and the total number of blocks.

A file system provides a user with blocks for data and inodes for files. Each file uses one inode, and file data is stored in a disk allocation unit (DAU). DAU sizes are determined at the time the file system is created. Quotas account for disk usage in multiples of 512 bytes.

The following sections provide background information about using quotas:

Table 64. shows the terms that are used extensively in this chapter's quota documentation.

Table 64. Quota Terminology

Term	Definition
grace period	The amount of time that can elapse during which a user is allowed to create files and/or allocate storage after users reach their soft limit.
soft limit	For disk quotas, a threshold limit on file system resources (blocks and inodes) that a user can temporarily exceed. Exceeding the soft limit starts a timer. When a user exceeds the soft limit for the specified time (the grace period), no further system resources can be allocated until the user reduces file system use below the soft limit.
hard limit	For disk quotas, a maximum limit on file system resources (blocks and inodes) that users cannot exceed.
quota	The amount of system resources that a user is allowed to consume.
timer	A facility for tracking the time elapsed after a user reaches a soft limit. When it reaches the grace period, a hard limit is imposed on the user.

Types of Quotas, Quota Files, and Quota Records

You can set quotas according to user ID, group ID, or an administrator's site-specific grouping. This site-specific grouping is called an *admin set ID*. You can use an admin set ID, for example, to identify a collection of users working on a project for which file system quotas are imposed.

Quotas are enabled when the system detects the presence of one or more quota files in the file system's root directory *and* the `quota` mount option is in effect. Note that the `quota` mount option is enabled by default, so you must not have disabled quotas by specifying the `noquota` mount option. If you mount the file system with `noquota` in effect, quotas are disabled. For more information about mount options, see the `mount_samfs`(1M) man page.

Each quota file contains a sequence of records. Record zero is the record for the system administrator's quotas. The system administrator's resource usage is accumulated in record zero. System administrator quotas are never enforced, but you can use any record, including the system administrator's record, as a template for subsequent records in the quota file. For more information about this practice, see "To Enable or Change Limits for Users, Groups, or Admin Sets Using an Existing Quota File" on page 211.

Record one is the record in the quota file for user one, group one, or admin set ID one, depending on the type of quota file. You can edit record one and all subsequent records in order to set different quotas for different users. Table 65. shows the quota file names and the quotas they enable in `/root`.

Table 65. Quota File Names

Quota File Name in `/root` Directory	Quota Type
`.quota_u`	UID (system user ID)
`.quota_g`	GID (system group ID)
`.quota_a`	AID (system admin set ID)

You can set default quota limits for users by editing record zero in the quota file and allowing the values in record zero to be used as the initial quota settings for all other users. By default, if user quota limits have not been set specifically, the system uses the values in record zero.

Soft Limits and Hard Limits

You can set both soft and hard limits. A *hard limit* specifies a fixed amount of system resources available for use, and the system never allows a user to exceed this limit. A *soft limit* specifies a level of system resource use that can be exceeded temporarily. The soft limits are never larger than the hard limits. If a new user attempts to allocate resources beyond his or her hard limit, the operation is aborted. In this case, the operation (typically a `write`(2) or `creat`(2)) fails and generates an `EDQUOT` error.

After a user exceeds a soft limit, a timer starts, and the user enters a grace period. While the timer is ticking, the user is allowed to operate above the soft limit but cannot exceed the hard limit. After the user goes below the soft limit, the timer is reset. If the grace period ends and the timer stops without the user having gone below the soft limit, the soft limit is then enforced as a hard limit.

For example, assume that a user has a soft limit of 10,000 blocks and a hard limit of 12,000 blocks. If the user's block usage exceeds 10,000 blocks and the timer exceeds the grace period, this user is no longer able to allocate more disk blocks on that file system until their usage drops below the 10,000-block soft limit.

You, the administrator, can use the `samquota`(1M) command to see the timer value. The `squota`(1) command is a user version of the `samquota`(1M) command. The `squota`(1) user command contains options that a user can specify to obtain information about quotas that pertain to them.

Quotas and Archive Media

You can use quotas to limit the amount of data that a user is allowed to have on archive media in StorageTek QFS and StorageTek ASM file systems.

Example. The `stage`(1) command brings data online from archive media. It is possible for a user quota to be exceeded when the `stage`(1) command is invoked at the system level in the following way:

```
# stage -r *
```

A user quota is observed when a user issues the `stage`(1) `-w` command, as follows:

```
# stage -w *
```

The system stages files until the user's quota is met. After that time, no more files are staged.

Disk Blocks and File Limits

It is possible for a user to exceed an inode quota, without using any blocks, by creating all empty files. It is also possible for a user to use only one inode and still exceed the block quota by creating a file that is large enough to consume all data blocks in the user's quota.

File system quotas are expressed in terms of the number of 512-byte blocks that a user can allocate. However, disk space is allocated to user files in terms of DAUs. The DAU setting is specified by the `-a` *allocation_unit* option to the `sammkfs`(1M) command. It is preferable to set a block quota to a multiple of the file system DAU. If this is not done, users can allocate only up to the block count, rounded down to the nearest DAU.

Enabling Quotas

You can enable quotas through a process that includes editing system files, creating quota files, and entering various quota commands.

Table 66. shows the commands used when manipulating quotas.

Table 66. Quota Commands

Command	Description
`squota`(1)	Displays quota statistics for a user. This is a subset of the `samquota`(1M) command.

Table 66. Quota Commands (Continued)

Command	Description
`samchaid`(1M)	Changes file admin set ID attributes.
`samquota`(1M)	Displays quota statistics for a user, group, or admin set. This command also enables an administrator to edit quota records.
`samquotastat`(1M)	Reports which, if any, quotas are active on a file system.

When it is run, the `samfsck`(1M) command checks the file system to make sure that usage values recorded in the quota files match the actual file system usage totals. If they do not match, the `samfsck`(1M) command issues notices, and it updates all existing, incorrect quota records if a file system repair is performed.

The following sections provide more details on how to configure a file system to use quotas and how to enable quotas.

Guidelines for Setting Up Quotas

Before you enable quotas, you should determine how much disk space and how many inodes to allocate to each user. If you want to be sure that the total file system space is never exceeded, you can divide the total size of the file system between the number of users. For example, if three users share a 100-megabyte slice and have equal disk space needs, you could allocate 33 megabytes to each. In environments in which not all users are likely to push their limits, you might want to set individual quotas so that they add up to more than the total size of the file system. For example, if three users share a 100-megabyte slice, you could allocate 40 megabytes to each.

You can use the following quota commands, in the formats shown, for displaying quota information:

- The `squota`(1) command is for end users. It enables them to retrieve quota information for themselves on a user, group, or admin set basis.
- The `samquota`(1M) command is for system administrators. It enables you to retrieve quota information or to set quotas. The `-U`, `-G`, and `-A` options on the `samquota`(1M) command determine whether the command is being used for a user, a group, or an admin set. Figure 102. shows this.

Figure 102. Using `samquota`(1M) to Retrieve Information

```
# samquota -U janet /mount_point    #Prints a user quota
# samquota -G pubs /mount_point     #Prints a group quota
# samquota -A 99 /mount_point       #Prints an admin set quota
```

To Configure a New File System to Use Quotas

The following procedure shows how to configure a new file system to use quotas. This procedure applies if you are creating a new file system at this time and no files currently reside in the file system.

To configure an existing file system to use quotas, see "To Configure an Existing File System to Use Quotas" on page 206.

1. Become superuser.
2. Create the file system.

   To create the file system, either follow the steps outlined in the *StorageTek ASM Installation and Configuration Guide* or use the examples in "Configuration Examples" on page 44 to guide you through creating the `mcf` file, creating the mount point, initializing the file system, and so on.
3. Use the `mount`(1M) command to mount the file system.

   Mount the file system using the `mount`(1M) command, as follows:

```
# mount /qfs1
```

4. Use the `dd`(1M) command to create the quota file(s).

   The arguments to this command differ depending on the type of quota you are creating, as follows:

   - To create admin set quotas, use the following command:

```
# dd if=/dev/zero of=/qfs1/.quota_a bs=4096 count=1
```

   - To create group quotas, use the following command:

```
# dd if=/dev/zero of=/qfs1/.quota_g bs=4096 count=1
```

   - To create user quotas, use the following command:

```
# dd if=/dev/zero of=/qfs1/.quota_u bs=4096 count=1
```

   For more information about the `dd`(1M) command, see the `dd`(1M) man page.
5. Use the `umount`(1M) command to unmount the file system.

Unmount the file system in which the quota files have been created using the `umount`(1M) command. For example:

```
# umount /qfs1
```

The file system needs to be unmounted so it can be remounted and have its quota files read at mount time. For more information about the `umount`(1M) command, see the `umount`(1M) man page.

6. Use the `samfsck`(1M) command to perform a file system check.

   Run the `samfsck`(1M) command on the file system. For example, the following command performs a file system check. The `-F` option resets the in-use values in the quota files.

```
# samfsck -F qfs1
```

7. Use the `mount`(1M) command to remount the file system.

   Quotas are enabled when the system detects the presence of one or more quota files in the root directory of a file system.

CAUTION: You do not need to include the quota mount option in the `/etc/vfstab` or `samfs.cmd` file. The `quota` mount option is enabled by default on the `mount`(1M) command, and quotas are enabled automatically when the system detects the presence of quota files. Make sure that you do not have the `noquota` mount option specified in your `samfs.cmd` or `/etc/vfstab` files.

If quota files are present and if the file system is mounted without quotas enabled, when blocks or files are allocated or freed, the quota records become inconsistent with actual usages. If a file system with quotas is mounted and run without the `quota` mount option, run `samfsck`(1M) with its `-F` option to update the quota file usage counts before again remounting the file system with quotas enabled.

For more information about the `mount`(1M) command, see the `mount_samfs`(1M) man page.

8. Use the `samquota`(1M) command to set quotas for users, groups, or admin sets.

   Subsequent sections in this chapter provide procedures and show examples of this process. For more information about the `samquota`(1M) command, see the `samquota`(1M) man page.

To Configure an Existing File System to Use Quotas

This procedure applies if you are creating quotas for a file system that is already populated with files.

If you are configuring a new file system to use quotas, see "To Configure a New File System to Use Quotas" on page 204.

1. Use the `su`(1) command to become superuser.
2. Use the `mount`(1M) command to ensure that the file system is mounted.

   Examine the `/etc/mnttab` file using the `mount`(1M) command with no arguments, as follows:

```
# mount
```

3. Use the `cd`(1) command to change to the root directory.

   Change to the root directory of the file system for which quotas are to be enabled. For example:

```
# cd /oldfs1
```

4. Verify that quotas do not already exist on the file system.

   From the root directory, use the `ls`(1) `-a` command to retrieve the list of files in this directory. If at least one quota type (`u`, `g`, or `a`) is established for a file system, you can establish any other quota type later. Be careful not to modify existing quota files when adding new ones later.

   If any of the following files are present, quotas are, or previously have been, enabled for this file system: `.quota_u`, `.quota_g`, `.quota_a`.

5. Use the `dd`(1M) command to create the quota file(s).

   Create the quota files for the type(s) of quota(s) you wish to enforce. Determine the highest existing ID numbers of the types of quotas you wish to enforce. Make the initial, zero quota files large enough to hold the record for those IDs; each quota file record requires 128 bytes.

   Example 1. If you want to enable admin set quotas, and the highest admin set ID in use on the file system is 1024, the calculation is as follows:

   - (1024+1)*128 = 131200
   - 131200/4096 = 32.031...

   Use the following command:

```
# dd if=/dev/zero of=/oldfs1/.quota_a bs=4096 count=33
```

Example 2. If you want to enable group quotas, and group IDs up to 2000 are in use, the calculation is as follows:

- (2000+1)*128 = 256128
- 256128/4096 = 62.531...

Use the following command:

```
# dd if=/dev/zero of=/oldfs1/.quota_g bs=4096 count=63
```

Example 3. If you want to enable user ID quotas, and user IDs up to 4799 are in use, the calculation is as follows:

- (4799+1)*128 = 1228800
- 1228800/4096 = 300.0

Use the following command:

```
# dd if=/dev/zero of=/oldfs1/.quota_u bs=4096 count=300
```

For more information about the `dd`(1M) command, see the `dd`(1M) man page.

6. Use the `umount`(1M) command to unmount the file system in which the quota files have been created.

   For example:

```
# umount /oldfs1
```

   The file system needs to be unmounted so it can be remounted and have its quota files read at mount time. For more information about unmounting a file system, see “Unmounting a File System” on page 64.

7. Use the `samfsck`(1M) command to perform a file system check.

   Use the `samfsck`(1M) `-F` command to perform a file system check. The `samfsck`(1M) command updates the quota files with correct, current usage information.

   For example:

```
# samfsck -F /oldfs1
```

Note: The command in this step updates only the records already allocated in the quota files.

8. Use the `mount`(1M) command to remount the file system in which the quota files have been created.

The system enables quotas when it detects the presence of one or more quota files in the /root directory.

CAUTION: You do not need to include the quota mount option in the /etc/vfstab or samfs.cmd file. The quota mount option is enabled by default on the mount(1M) command, and quotas are enabled automatically when the system detects the presence of quota files. Make sure that you do not have the noquota mount option specified in your samfs.cmd or /etc/vfstab files.

If quota files are present and if the file system is mounted without quotas enabled, when blocks or files are allocated or freed, the quota records become inconsistent with actual usages. If a file system with quotas is mounted and run without the quota mount option, run the samfsck(1M) -F command to update the quota file usage counts before again remounting the file system with quotas enabled.

For more information about the mount(1M) command, see the mount_samfs(1M) man page.

9. Use the samquota(1M) command to set quotas for users, groups, or admin sets.

   Subsequent sections in this chapter provide procedures and show examples of this process. For more information about the samquota(1M) command, see the samquota(1M) man page.

To Assign Admin Set IDs to Directories and Files

1. Use the su(1) command to become superuser.
2. Set the admin IDs.

   Use the samchaid(1M) command to change the admin set IDs for the directory or file, as follows:

   - To set IDs for a file or directory, specify the directory name or path. For example:

```
# samchaid 100 admin.dir
```

- To set IDs for a directory tree, use the -R and (if necessary) the -h options. The -R option specifies a recursive operation, and the -h option changes links, not targets. For example:

```
# samchaid -R -h 22 /qfs1/joe /qfs1/nancee
```

For more information about the `samchaid`(1M) command, see the `samchaid`(1M) man page.

Setting Infinite Quotas

An *infinite quota* is a kind of special quota. Users with infinite quotas are never denied access to any available file system resource. You can set infinite quotas on a user, group, or admin set basis by setting both the hard block and hard file limits to zero. The file system treats an infinite quota as a special quota. You can set infinite quota values into record zero of the user, group, or admin set ID quota files, and from there they can become the default values for new users, groups, or admin set IDs.

To Set an Infinite Quota

1. Use the `samquota`(1M) command to set an infinite quota.

   For example, the following command sets an infinite quota:

```
# samquota -U fred -b 0:h -f 0:h /qfs1
```

   You can use the `samquota`(1M) command to set infinite quotas for particular users, groups, or admin set IDs by setting zero values for all hard and soft limits. Figure 103. shows how to set infinite quotas.

Figure 103. Setting Infinite Quotas

```
# samquota -G sam -b 0:s,h -f 0:s,h /sam6
# samquota -G sam /sam6



                                 Online Limits                Total Limits
        Type    ID    In Use    Soft      Hard    In Use     Soft      Hard
/sam6
Files  group   101       339       0         0       339        0         0
Blocks group   101       248       0         0      2614        0         0
Grace period                    0s                           0s
---> Infinite quotas in effect.
```

Enabling Default Quota Values

You can use the `samquota`(1M) command to enable a default quota for a user, group, or admin set. This is accomplished by setting default limits into user, group, or admin set zero (0).

To Enable Default Quota Values for Users, Groups, or Admin Sets

1. Use the samquota(1M) command to set an infinite quota.

   For example, the following samquota(1M) command sets default quotas for all admin set IDs:

```
# samquota -A 0 -b 12000:s -b 15000:h -b 12G:s:t -b 15G:h:t \
-f 1000:s -f 1200:h -t 1w /qfs1
```

   On first reference, the preceding command sets any user's uninitialized admin set quota limits as follows:

   - The soft online block limit is set to 12,000 blocks.
   - The hard online block limit is set to 15,000 blocks.
   - The total soft block limit is set to 12 gigablocks.
   - The total hard block limit is set to 15 gigablocks.
   - The soft file limit is set to 1000 files.
   - The hard file limit is set to 1200 files.
   - The grace period is set to one week.

   Note that if a quota record already exists, the existing values remain in effect. This occurs, for example, if the admin group already has blocks assigned to it.

   You can set similar default quotas for users or groups by specifying `-U 0` or `-G 0`, respectively, in place of `-A 0`.

   For more information about the samquota(1M) command, see the samquota(1M) man page.

Enabling Limits

You can use the samquota(1M) command to enable a set of limits for a particular user, group, or admin set.

To Enable Limits for Users, Groups, or Admin Sets

1. Use the `samquota`(1M) command to set limits for users, groups, or admin sets.

   Figure 104. shows commands that enable various limits.

Figure 104. Quota Commands

```
# samquota -U joe -b 15000:s -b 20000:h -b 12G:s:t -b 15G:h:t \
-f 500:s -f 750:h -t 3d /qfs1
# samquota -G proj -b 15000:s -b 20000:h -b 12G:s:t -b 15G:h:t \
-f 500:s -f 750:h -t 3d /qfs1
# samquota -A 7 -b 15000:s -b 20000:h -b 12G:s:t -b 15G:h:t \
-f 500:s -f 750:h -t 3d /qfs1
```

   For more information about the `samquota`(1M) command, see the `samquota`(1M) man page.

To Enable or Change Limits for Users, Groups, or Admin Sets Using an Existing Quota File

After quotas are established, you can use an existing quota file as a template when creating limits for another user, group, or admin set. The following procedure shows this. You can also use this procedure to change any of the quota settings.

1. Use the `samquota`(1M) command to retrieve a quota file.

   Use the `-e` option with one or more of the following additional options: `-U` *userID*, `-G` *groupID*, or `-A` *adminsetID*. Direct the output to a temporary file.

```
# samquota -G sam -f 200:s:o   -f 300:h:o   -f 200:s:t -f 300:h:t \
-b 40000:s:o -b 60000:h:o -b 40M:s:t -b 60M:h:t -t 0s:o -t 0s:t /sam6
```

   You can use any temporary file. In Step 2, you use an editor to change one or more fields, so you can use a group quota entry as a template to create a user quota entry. Figure 105. shows how to create and retrieve file `quota.group` to use as a template.

Figure 105. File `quota.group`

```
# samquota -G sam -e /sam6 > /tmp/quota.group
# cat /tmp/quota.group

# Type  ID
#                Online Limits                              Total   Limits
#            soft            hard                       soft             hard
# Files
# Blocks
# Grace Periods
#
samquota -G 101 \
      -f      200:s:o -f      300:h:o         -f      200:s:t -f
300:h:t \
      -b    40000:s:o -b    60000:h:o         -b 40000000:s:t -b
60000000:h:t \
                 -t  0s:o                                -t 0s:t   /sam6
```

2. Use an editor to edit the file from Step 1.

   For example, Figure 106. shows the file that was generated in Step 1 opened in the `vi`(1) editor. This file also shows that group ID `101` is changed to `102`. This has the effect of generating a command to copy the quotas set for group `101` to group `102`.

Figure 106. File `quota.group` **After Editing**

```
# Type  ID
#                Online Limits                              Total   Limits
#            soft            hard                       soft             hard
# Files
# Blocks
# Grace Periods
#
samquota -G 102 \
      -f      200:s:o -f      300:h:o         -f      200:s:t -f
300:h:t \
      -b    40000:s:o -b    60000:h:o         -b 40000000:s:t -b
60000000:h:t \
                 -t  1d:o                                -t 1d:t   /sam6
```

3. Save the file and exit the editor.
4. Execute the file using the shell.

   This step applies the changes made in the editor. For example:

```
# sh -x /tmp/quota.group
```

In this example, the `-x` option directs the shell to echo the commands it executes. You can omit the `-x` option if desired.

In similar fashion, you can use this procedure to generate quota commands that copy quota limits between users, groups, admin IDs, file systems, and other entities.

Checking Quotas

After you have enabled disk and inode quotas, you can check these quotas. The `samquota`(1M) command is an administrator command that generates a quota report on an individual user, group, or admin set. The `squota`(1) command is a user command that enables users to check their own individual quotas. Table 67. shows commands you can use to check quotas.

Table 67. Commands for Checking Quotas

Command	Task
`squota`(1)	This is a user command. It displays user quotas and other information specific to a single user. For more information, see the `squota`(1) man page.
`samquota`(1M)	This is an administrator command. It displays user, group, and admin set quotas, and it displays current disk use. This command also displays information about users who are exceeding their quotas. For more information, see the `samquota`(1M) man page.

To Check for Exceeded Quotas

The following procedure shows how to check quotas for excess usage.

1. Become superuser.
2. Use the `samquota`(1M) command to display the quotas in effect.

   Use the `samquota`(1M) command in one of the following ways to display quotas for mounted file systems in which quotas are enabled:

- To display user quotas, specify the following command:

```
# samquota -U userID [ file ]
```

For *userID*, specify the numeric user ID or user name of the user whose quotas are being examined.

For *file*, specify a specific file system for the selected user, group, or admin set. The *file* argument can also be the name of any file in the file system. Typically, file is the name of the root directory of the file system.

Example 1. Figure 107. retrieves user `hm1259`'s quota statistics in the `sam6` file system on the server and displays output indicating that this user is not exceeding his quota.

Figure 107. Checking for Exceeded Quotas for User `hm1259`

```
# samquota -U hm1259  /sam6

                                  Online Limits                 Total  Limits
        Type    ID     In Use     Soft      Hard     In Use     Soft      Hard
/sam6
Files   user 130959        13      100       200         13      100       200
Blocks  user 130959       152      200      3000        272     1000      3000
Grace period                      0s                            0s
```

Example 2. Figure 108. retrieves user `memil`'s quota statistics in all mounted StorageTek QFS and StorageTek ASM file systems and displays output indicating that this user is exceeding the quota. Note the plus sign (+) in the `Blocks` row of the output. The plus sign would appear in the `Files` row, too, if the soft quota limit were being exceeded for files.

Figure 108. Checking for Exceeded Quotas for User `memil`

```
# samquota -U memil

                                  Online Limits                 Total  Limits
        Type    ID     In Use     Soft      Hard     In Use     Soft      Hard
/sam6
Files   user 130967         4      500       750          4      500       750
Blocks  user 130967    41016+    40000     50000      41016    50000     50000
Grace period                      1w                            0s
---> Warning:  online soft limits to be enforced in 6d23h36m45s
/sam7
Files   user 130967         4      500       750          4      500       750
Blocks  user 130967      4106    40000     50000       4106    50000     50000
Grace period                      1w                            0s
```

If a hard limit has been exceeded, or if the soft limit has been exceeded and the grace period has expired, the offending `In Use` field is marked with an asterisk character (`*`). If a quota record's limits are determined to be inconsistent, (for example if a soft limit is larger than a hard limit), an exclamation point is used to mark the field, and all allocation operations are prevented.

Table 68. shows the fields in the `samquota`(1M) output.

Table 68. `samquota`(1M) Output Fields

Field Name	Content
`In Use`	Current block usage.
`Soft`	Soft block limit
`Hard`	Hard block limit
`Grace Period`	Amount of time the user is allowed to exceed the soft limit

- To display group quotas, specify the following command:

```
# samquota -G groupID [ file ]
```

  For *groupID*, specify the numeric group ID or the group name for the group of users whose quotas are being examined. For example, the following command retrieves user quota statistics for the group `turtles` in the `qfs3` file system:

```
# samquota -G turtles /qfs3
```

- To display admin set quotas, specify the following command:

```
# samquota -A adminsetID [ file ]
```

  For *adminsetID*, specify the numeric admin set ID of the site-specific administrator set whose quotas are being examined. For example, the following command retrieves user quota statistics for the admin set 457 in all mounted StorageTek QFS and StorageTek ASM file systems:

```
# samquota -A 457 /qfs3
```

Changing and Removing Quotas

You can change quotas to adjust the amount of disk space or number of inodes allocated to users. You can also remove quotas from users or from an entire file system. The following sections describe how to change and remove quotas. The topics are as follows:

- "To Change the Grace Period" on page 216
- "Changing the Grace Period Expiration" on page 218
- "To Inhibit Additional File System Resource Allocations" on page 220

- “To Remove a File System’s Quotas” on page 222
- “To Correct Quotas” on page 223

To Change the Grace Period

You can use the `samquota`(1M) command to change the soft time limit grace period.

1. Use the `samquota`(1M) command to retrieve quota statistics.

   You can use the `samquota`(1M) command on a user, group, or admin set basis. Figure 109. shows how to retrieve quota statistics.

Figure 109. Using `samquota`(1M) to Retrieve Quota Statistics

```
# samquota -U userID [ file ]
# samquota -G groupID [ file ]
# samquota -A adminsetID [ file ]
```

Table 69. shows the arguments to these commands.

Table 69. `samquota`(1M) Command Arguments

Argument	Description
userID	Specify the numeric user ID or user name of the user whose quotas are being changed.
groupID	Specify the numeric group ID or the group name for the group of users whose quotas are being changed.
adminsetID	Specify the numeric admin set ID of the site-specific administrator set whose quotas are being changed.
file	Specify a specific file system for the selected user, group, or admin set. The *file* argument can also be the name of any file in the file system. Typically, *file* is the name of the root directory of the file system.

2. Examine the output from the `samquota`(1M) command.

   Examine the output and determine what the new limits should be.

3. Use the `samquota`(1M) command to change the soft time limit grace period.

Figure 110. shows using the `samquota`(1M) command options to use to change the soft time limit grace period.

Figure 110. Using `samquota`(1M) to Change Soft Time Limit Grace Periods

```
# samquota -U userID -t interval file
# samquota -G groupID -t interval file
# samquota -A adminID -t interval file
```

Table 70. shows the arguments to these commands.

Table 70. `samquota`(1M) Command Arguments

Argument	Description
userID	Specify the numeric user ID or user name of the user whose quotas are being changed.
groupID	Specify the numeric group ID or the group name for the group of users whose quotas are being changed.
adminsetID	Specify the numeric admin set ID of the site-specific administrator set whose quotas are being changed.
interval	Specifies the interval to use for the grace period. Specify an integer number for *interval* to indicate the quantity, and then specify a unit multiplier, if desired. By default, the unit multiplier is `s` to indicate that the *interval* is being specified in seconds. You can also specify `w` (for weeks), `d` (for days), `h` (for hours), or `m` (for minutes).
file	Specify a specific file system for the selected user, group, or admin set. The *file* argument can also be the name of any file in the file system. Typically, *file* is the name of the root directory of the file system.

Example. Assume that you want to change the grace period for user `memil`. Figure 111. shows the `samquota`(1M) command used to verify the quotas and its output.

Figure 111. Changing the Grace Period

```
# samquota -U memil /sam6

                                Online Limits                  Total Limits
        Type    ID    In Use    Soft      Hard    In Use    Soft      Hard
/sam6
Files   user 130967        4     500       750         4     500       750
Blocks  user 130967   41016+   40000     50000     41016   50000     50000
Grace period                    3d                            0s
---> Warning:  online soft limits to be enforced in 2d23h59m7s
```

You enter the following command to lower the soft time limits:

```
# samquota -U memil -t 1d /sam6
```

Figure 112. shows the samquota(1M) command to use to verify the new quotas.

Figure 112. Verifying the New Quotas

```
# samquota -U memil /sam6
                                Online Limits                 Total  Limits
        Type    ID    In Use    Soft      Hard     In Use     Soft      Hard
/sam6
Files   user 130967         4     500      750          4      500       750
Blocks  user 130967    41016+   40000    50000      41016    50000     50000
Grace period                    1d                              0s
---> Warning:  online soft limits to be enforced in 23h58m31s
```

Changing the Grace Period Expiration

If a user has exceeded their soft quota limit, changing the grace period itself does not modify the expiration timer of any grace periods that have already started. If the grace period is already in effect, you can use the samquota(1M) command to modify the grace period in one of the following ways:

- **Clear the grace period.** The next time the user allocates a file or block (and is still over a soft limit), the grace period timer is reset to the grace period and starts counting down.
- **Reset the grace period.** When an expiration period is reset, the timer is reset to the present grace period, which starts counting down immediately.
- **Set the grace period to a value.** The timer is set to a value, and it starts counting down immediately from that value. There are no restrictions on this value. The value can be larger than the grace period.
- **Expire the grace period.** The timer is set to expire immediately.

Example. Figure 113. retrieves information about group sam and shows that this group is over its soft limit.

Figure 113. Exceeding a Soft Limit

```
# samquota -G sam  /sam6

                                Online Limits                 Total  Limits
        Type    ID    In Use    Soft      Hard     In Use     Soft      Hard
/sam6
Files   group   101       32     2000      2000        32     2000      2000
Blocks  group   101   41888*   40000 60000000     43208 60000000 60000000
Grace period                    1w                              1w
---> Online soft limits under enforcement (since 30s ago)
```

Figure 114. clears the timer so it starts counting the next time a user in group `sam` attempts to allocate a block or file in `/sam6`.

Figure 114. Clearing the Timer

```
# samquota -G sam -x clear /sam6
Setting Grace Timer:  continue? y
# samquota -G sam  /sam6

                                Online Limits                 Total  Limits
        Type    ID    In Use     Soft     Hard    In Use      Soft     Hard
/sam6
Files   group   101       32     2000     2000        32      2000     2000
Blocks group    101    41888+   40000 60000000     43208 60000000 60000000
Grace period                     1w                            1w
---> Warning:  online soft limits to be enforced in 6d23h59m56s
```

Figure 115. resets the grace period.

Figure 115. Resetting the Grace Period

```
# samquota -G sam -x reset /sam6
Setting Grace Timer:  continue? y
# samquota -G sam  /sam6
                                Online Limits                 Total  Limits
        Type    ID    In Use     Soft     Hard    In Use      Soft     Hard
/sam6
Files   group   101       32     2000     2000        32      2000     2000
Blocks group    101    41888    40000 60000000     43208 60000000 60000000
Grace period                     1w                            1w
---> Warning:  online soft limits to be enforced in 6d23h59m52s
```

Figure 116. expires the grace period.

Figure 116. Expiring the Grace Period

```
# samquota -G sam -x expire /sam6
Setting Grace Timer:  continue? y
# samquota -G sam  /sam6

                                Online Limits                 Total  Limits
        Type    ID    In Use     Soft     Hard    In Use      Soft     Hard
/sam6
Files   group   101       32     2000     2000        32      2000     2000
Blocks group    101    41888    40000 60000000     43208 60000000 60000000
Grace period                     1w                            1w
---> Online soft limits under enforcement (since 6s ago)
```

Figure 117. sets a very long expiration period.

Figure 117. Setting a Very Long Grace Period

```
# samquota -G sam -x 52w /sam6
Setting Grace Timer:  continue? y
# samquota -G sam  /sam6

                                 Online Limits                    Total  Limits
        Type    ID    In Use     Soft      Hard    In Use     Soft      Hard
/sam6
Files  group   101        32     2000      2000        32     2000      2000
Blocks group   101    41888+    40000 60000000    43208 60000000 60000000
Grace period                     1w                             1w
---> Warning:  online soft limits to be enforced in 51w6d23h59m54s
```

To Inhibit Additional File System Resource Allocations

When the file system detects that quota values are not consistent for a user, group, or admin set, it prevents that user, group, or admin set from using any more system resources. You can inhibit file system resource allocations by creating inconsistent quota values. The `samquota`(1M) command detects these inconsistent values, and reports them in its output. For example, the software inhibits further allocation if the hard block or file limits are lower than the soft block or file limits, or if a user's soft limit is larger than the user's hard limit.

The file system treats an inconsisent quota setting as a special quota. You can set inconsistent quota values into record zero of the user, group, or admin set ID quota files, and from there they can become the default values for new users, groups, or admin set IDs.

The following procedure shows how to inhibit further system resource allocations for a user, group, or admin set.

1. Become superuser.
2. Obtain, save, and examine current quota information.

   Figure 118. shows how to use the `samquota`(1M) command to retrieve current group quota information for group `sam` and write it to a backup file.

Figure 118. Retrieving Group Quota Information

```
# samquota -G sam -e /sam6 | & tee restore.quota.sam

# Type  ID
#                Online Limits                          Total   Limits
#            soft            hard                    soft             hard
# Files
# Blocks
# Grace Periods
```

Figure 118. Retrieving Group Quota Information (Continued)

```
#
samquota -G 101 \
      -f     2000:s:o -f     2000:h:o         -f     2000:s:t -f
2000:h:t \
      -b    40000:s:o -b 60000000:h:o         -b 60000000:s:t -b
60000000:h:t \
                 -t  1w:o                               -t 1w:t \
      -x 51w6d23h59m:o                       -x clear     /sam6
```

To obtain quota information about a user quota, specify the `-U` *userID* option in place of the `-G` option. To obtain quota information about an admin set quota, specify the `-A` *adminID* option in place of the `-G` option.

3. Use the `samquota`(1M) command to set soft quotas to nonzero quotas and hard quotas to zero quotas.

   Use the `samquota`(1M) command to reset the quotas to invalid values. The following command sets the quotas for group `sam` to be inconsistent:

```
# samquota -G sam -f 1:s -f 0:h -b 1:s -b 0:h /sam6
```

   To make the quotas for users or admin sets inconsistent, specify the `-U` *userID* or `-A` *adminID* options in place of the `-G` option.

4. Use the `samquota`(1M) command to verify your changes.

   Use the `samquota`(1M) command to verify that the quota has been correctly changed. The following example obtains quota information for a group quota of the group `sam`:

```
# samquota -G sam /qfs1
```

   Enter the `samquota`(1M) command again to verify the changed quotas. Figure 119. shows this.

Figure 119. Verifying Changed Quotas

```
# samquota -G sam  /sam6

                                Online Limits                Total  Limits
        Type     ID     In Use     Soft     Hard     In Use     Soft     Hard
/sam6
Files   group   101        32!        1        0        32!        1        0
Blocks group   101     41888!        1        0     43208!        1        0
Grace period                    1w                          1w
---> Quota values inconsistent; zero quotas in effect.
```

In the preceding output, a zero quota is in effect. Note the exclamation point characters (`!`) to indicate the over-quota condition in the output.

5. Use the `sh`(1) and `samquota`(1M) commands to restore the group's quota.

   Figure 120. shows the commands to restore and verify the changed quotas.

Figure 120. Restoring the Group Quota

```
# sh restore.quota.sam
Setting Grace Timer:  continue? y
Setting Grace Timer:  continue? y
# samquota -G sam  /sam6
                              Online Limits                 Total Limits
        Type    ID    In Use    Soft     Hard    In Use     Soft     Hard
/sam6
Files  group   101        32     2000     2000        32     2000     2000
Blocks group   101    41888+   40000 60000000     43208 60000000 60000000
Grace period                    1w                          1w
---> Warning:  online soft limits to be enforced in 6d23h59m54s
```

To perform this operation on a user quota, specify the `-U` *userID* option in place of the `-G` option. To perform this operation on an admin set quota, specify the `-A` *adminID* option in place of the `-G` option.

To Remove a File System's Quotas

To remove or disable quotas for a file system, you need to remove quota specifications from the mount process. The following procedure shows how to disable quotas for a file system.

1. Use the `su`(1) command to become superuser.

2. Add the `noquota` mount option to the `/etc/vfstab` or `samfs.cmd` file. (Optional)

   Perform this step only if you have the `quota` mount option in the `/etc/vfstab` or `samfs.cmd` file.

   Use a viewer, such as `vi`(1) or `cat`(1) to examine the `/etc/vfstab` or `samfs.cmd` file for the presence of the `quota` mount option.

   If this mount option is present, edit the file and remove the `quota` mount option.

Note: Beginning with the StorageTek QFS and StorageTek ASM 4.1 releases, you do not need to use the `/etc/vfstab` and `samfs.cmd` files for enabling or disabling quotas.

3. Use the `umount`(1M) command to unmount the file system.

   If the file system is mounted, use the `umount`(1M) command to unmount the file system.

For example:

```
# umount /myfs
```

If you have difficulty unmounting the file system, see "Unmounting a File System" on page 64.

4. Remount the file system using the mount(1M) command.

   If you did not perform Step 2, include the noquota option on the mount(1M) command.

   For example:

```
# mount -o noquota /myfs
```

5. Dispense with the quota files.

   If you expect to reinstate the quota feature at a later date, do not destroy the quota files. To preserve the quota files and reinstate quotas at a later date, unmount the file system, run the samfsck(1M) command with its -F option on the file system, and remount the file system again with the quota mount option. The quota mount option can be specified in either the /etc/vfstab file or in the samfs.cmd file as a mount option, or it can be specified on the mount(1M) command with the -o quota option.

   If you do not expect to reinstate the quota feature at a later date, or if you want to reclaim the space consumed by the quota files, use the rm(1) command to remove the .quota_u, .quota_g, and .quota_a files. For example:

```
# rm /myfs/.quota_[agu]
```

To Correct Quotas

1. Become superuser.
2. Use the umount(1M) command to unmount the file system.

   If the file system is mounted, use the umount(1M) command to unmount the file system.

   For example:

```
# umount /myfs
```

If you have difficulty unmounting the file system, see "Unmounting a File System" on page 64.

3. Use the `samfsck`(1M) `-F` command to perform a file system check.

   The `samfsck`(1M) command updates the quota files with correct, current usage information. Note, however, that it updates only records already allocated in the quota files. For example:

```
# samfsck -F myfs
```

4. Use the `mount`(1M) command to remount the file system.

   For example:

```
# mount /myfs
```

StorageTek QFS in a Sun Cluster Environment

8

This chapter describes how the StorageTek QFS software works in a Sun Cluster environment. It also provides configuration examples for a StorageTek QFS shared file system in a Sun Cluster environment and for an unshared StorageTek QFS file system in a Sun Cluster environment.

This chapter contains the following sections:

Before You Begin

With version 4.2 of the StorageTek QFS software, you can install a StorageTek QFS file system in a Sun Cluster environment and can configure the file system for high availability. The configuration method you use varies, depending on whether your file system is shared or unshared.

This chapter assumes that you are an experienced user of both the StorageTek QFS software and the Sun Cluster environment. It also assumes you have performed either or both of the following:

- You have configured file systems as highly available scalable or failover resources under Sun Cluster control.
- You have installed and configured both StorageTek QFS and StorageTek QFS shared file systems.

It is recommended that you read the following documentation before continuing with this chapter:

- The following chapters of the *StorageTek ASM Installation and Configuration Guide*:
  - Chapter 1, which provides hardware and software prerequisite information for the StorageTek QFS software in a Sun Cluster environment.
  - Chapter 3, which describes how to install a StorageTek QFS file system on a Sun Cluster system for high availability.
- The following Sun Cluster documentation:
  - *Sun Cluster Concepts Guide for Solaris OS*, which introduces the Sun Cluster environment. In particular, please review the following sections in the guide:

  Local Disks

  Global Devices

  Device ID (DID)

  Disk Device Groups

  Disk Device Group Failover

  Local and Global Namespaces

  Cluster File Systems

  `HAStoragePlus` Resource Type

  Volume Managers

  - The *Sun Cluster Software Installation Guide for Solaris OS*, which describes the steps for installing the Sun Cluster software.
  - The *Sun Cluster Data Services Planning and Administration Guide for Solaris OS*, which explains how to plan your Sun Cluster environment to include various data services.
  - The *Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS*, which describes how to use the StorageTek QFS shared file system with the Sun Cluster data service for Oracle Real Application Clusters.

Note: All references in this document to “Oracle Real Application Clusters” apply also to “Oracle Parallel Server” unless otherwise specified.

■ Restrictions

The following restrictions apply to the StorageTek QFS software in a Sun Cluster environment:

- StorageTek ASM software, that is, use of Application Storage Manger (ASM) configurations with StorageTek QFS—is not supported.
- The following apply to shared file systems:
  - They can be configured only under the `SUNW.qfs` resource type. They cannot be configured under the `HAStoragePlus` resource type.
  - They can be configured as scalable file systems used only by the Sun Cluster data service for Oracle Real Application Clusters.
  - Shared file system data cannot be accessed from a node outside the Sun Cluster. This is true even if the node outside the Sun Cluster is configured only as a shared StorageTek QFS reader. To access file system data, all nodes must be members of the same Sun Cluster.
- The following apply to unshared file systems:
  - They can be configured only under the `HAStoragePlus` resource type. They cannot be configured under the `SUNW.qfs` resource type.
  - Any failover application that is supported with Sun Cluster software is supported with unshared StorageTek QFS software as a highly available local file system. Support is not limited to specific applications.
  - An unshared StorageTek QFS file system cannot be the underlying native file system in a Sun Cluster.

How the Sun Cluster and the StorageTek QFS Software Interact

The shared file system uses Sun Cluster Disk ID (DID) support to enable data access by the Sun Cluster data service for Oracle Real Application Clusters. The unshared file system uses global device volume support and volume manager-controlled volume support to enable data access by failover applications supported by Sun Cluster.

Data Access With a Shared File System

With DID support, each device that is under the control of the Sun Cluster system, whether it is multipathed or not, is assigned a unique disk ID. For every unique DID device, there is a corresponding global device. The StorageTek QFS shared file system can be configured on redundant storage that consists only of DID devices (`/dev/did/*`), where DID devices are accessible only on nodes that have a direct connection to the device through a host bust adapater (HBA).

Configuring the StorageTek QFS shared file system on DID devices and configuring the `SUNW.qfs` resource type for use with the file system makes the file system’s shared metadata server highly available. The Sun Cluster data service for Oracle Real Application Clusters can then access data from within the file system. Additionally, the StorageTek QFS Sun Cluster agent can then automatically relocate the metadata server for the file system as necessary.

Data Access With an Unshared File System

A global device is Sun Cluster’s mechanism for accessing an underlying DID device from any node within the Sun Cluster, assuming that the nodes hosting the DID device are available. Global devices and volume manager-controlled volumes can be made accessible from every node in the Sun Cluster. The unshared StorageTek QFS file system can be configured on redundant storage that consists of either raw global devices (`/dev/global/*`) or volume manager-controlled volumes.

Configuring the unshared file system on these global devices or volume manager-controlled devices and configuring the `HAStoragePlus` resource type for use with the file system makes the file system highly available with the ability to fail over to other nodes.

■ About Configuration Examples

This chapter provides configuration examples for the StorageTek QFS shared file system on a Sun Cluster and for the unshared StorageTek QFS file system on a Sun Cluster. All configuration examples are based on a platform consisting of the following:

- A two-node SunPlex™ system (`scnode-A` and `scnode-B`)
- DID devices that contain multi-host-accessible disk devices for both high availability and redundancy

All configurations in this chapter are also based on Figure 121.. In this code example, the `scdidadm(1M)` command displays the disk identifier (DID) devices, and the `-L` option lists the DID device paths, including those on all nodes in the Sun Cluster system.

Figure 121. Command That Lists the DID Devices and Their DID Device Paths

```
# scdidadm -L
1   scnode-A:/dev/dsk/c0t0d0    /dev/did/dsk/d1
2   scnode-A:/dev/dsk/c0t1d0    /dev/did/dsk/d2
3   scnode-A:/dev/dsk/c0t6d0    /dev/did/dsk/d3
4   scnode-A:/dev/dsk/c6t1d0    /dev/did/dsk/d4
4   scnode-B:/dev/dsk/c7t1d0    /dev/did/dsk/d4
5   scnode-A:/dev/dsk/c6t2d0    /dev/did/dsk/d5
5   scnode-B:/dev/dsk/c7t2d0    /dev/did/dsk/d5
6   scnode-A:/dev/dsk/c6t3d0    /dev/did/dsk/d6
6   scnode-B:/dev/dsk/c7t3d0    /dev/did/dsk/d6
7   scnode-A:/dev/dsk/c6t4d0    /dev/did/dsk/d7
7   scnode-B:/dev/dsk/c7t4d0    /dev/did/dsk/d7
8   scnode-A:/dev/dsk/c6t5d0    /dev/did/dsk/d8
8   scnode-B:/dev/dsk/c7t5d0    /dev/did/dsk/d8
9   scnode-B:/dev/dsk/c0t6d0    /dev/did/dsk/d9
10  scnode-B:/dev/dsk/c1t0d0    /dev/did/dsk/d10
11  scnode-B:/dev/dsk/c1t1d0    /dev/did/dsk/d11
```

Figure 121. shows that DID devices `d4` through `d8` are accessible from both Sun Cluster systems (`scnode-A` and `scnode-B`). With the StorageTek QFS file system sizing requirements and with knowledge of your intended application and configuration, you can decide on the most appropriate apportioning of devices to file systems. By using the Solaris `format(1M)` command, you can determine the sizing and partition layout of each DID device and resize the partitions on each DID device, if needed. Given the available DID devices, you can also configure multiple devices and their associated partitions to contain the file systems, according to your sizing requirements.

Configuring a StorageTek QFS Shared File System on a Sun Cluster

When you install a StorageTek QFS shared file system on a Sun Cluster, you configure the file system's metadata server under the `SUNW.qfs` resource type. This makes the metadata server highly available and enables the StorageTek QFS shared file system to be globally accessible on all configured nodes in the Sun Cluster.

A StorageTek QFS shared file system is typically associated with a scalable application. The StorageTek QFS shared file system is mounted on, and the scalable application is active on, one or more Sun Cluster nodes.

If a node in the Sun Cluster system fails, or if you switch over the resource group, the metadata server resource (StorageTek QFS Sun Cluster agent) automatically relocates the file system's metadata server as necessary. This ensures that the other nodes' access to the shared file system is not affected.

Note: To manually relocate the metadata server for a StorageTek QFS shared file system that is under Sun Cluster control, you must use the Sun Cluster administrative commands. For more information about these commands, see the Sun Cluster documentation.

Metadata Server Resource Considerations

When the Sun Cluster boots, the metadata server resource ensures that the file system is mounted on all nodes that are part of the resource group. However, the file system mount on those nodes is not monitored. Therefore, in certain failure cases, the file system might be unavailable on certain nodes, even if the metadata server resource is in the online state.

If you use Sun Cluster administrative commands to bring the metadata server resource group offline, the file system under the metadata server resource remains mounted on the nodes. To unmount the file system (with the exception of a node that is shut down), you must bring the metadata server resource group into the unmanaged state by using the appropriate Sun Cluster administrative command.

To remount the file system at a later time, you must bring the resource group into a managed state and then into an online state.

Example Configuration

This section shows an example of the StorageTek QFS shared file system installed on raw DID devices with the Sun Cluster data service for Oracle Real Application Clusters. For detailed information on how to use the StorageTek QFS shared file system with the Sun Cluster data service for Oracle Real Application Clusters, see the *Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS*.

As shown in Figure 121., DID devices `d4` through `d8` are highly available and are contained on controller-based storage. For you to configure a StorageTek QFS shared file system on a Sun Cluster, the controller-based storage must support device redundancy by using RAID-1 or RAID-5.

For simplicity in this example, two file systems are created:

- `qfs1`—This file system is used for the Oracle Real Application Clusters shared installation, configuration, and log files.

- `qfs2`—This file system is used for the database files that are shared by the Oracle Real Application Clusters software.

Additionally, device `d4` is used for StorageTek QFS metadata. This device has two 50 GB slices. The remaining devices, `d5` through `d8`, are used for StorageTek QFS file data.

This configuration involves five main steps, as detailed in the following subsections:

1. Preparing to create StorageTek QFS file systems.
2. Creating the file systems and configuring the Sun Cluster nodes.
3. Validating the configuration.
4. Configuring the network name service.
5. Configuring the Sun Cluster data service for Oracl Real Application Clusters.

To Prepare to Create StorageTek QFS Shared File Systems

Steps 1 through 3 in this procedure must be performed from one node in the Sun Cluster system. In this example, the steps are performed from node `scnode-A`.

1. From one node in the Sun Cluster system, use the `format`(1M) utility to lay out partitions on `/dev/did/dsk/d4`.

Figure 122. Laying Out Partitions on `/dev/did/dsk/d4`

```
# format /dev/did/rdsk/d4s2
# format> partition
[ output deleted ]
# partition> print
Current partition table (unnamed):
Total disk cylinders available: 12800 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size           Blocks
  0        usr    wm       1 -  6400       50.00GB    (6400/0/0)
104857600
  1        usr    wm    6401 - 12800       50.00GB    (6400/0/0)
104857600
  2     backup    wu       0 - 12800      100.00GB    (6400/0/0)
209715200
  3 unassigned   wu      0               0        (0/0/0)           0
  4 unassigned   wu      0               0        (0/0/0)           0
  5 unassigned   wu      0               0        (0/0/0)           0
  6 unassigned   wu      0               0        (0/0/0)           0
  7 unassigned   wu      0               0        (0/0/0)           0

NOTE: Partition 2 (backup) will not be used and was created by
format(1M) by default.
```

Partition (or slice) 0 skips over the volume's Volume Table of Contents (VTOC) and is then configured as a 50 GB partition. Partition 1 is configured to be the same size as partition 0.

2. Use the `format`(1M) utility to lay out partitions on `/dev/did/dsk/d5`.

Figure 123. Laying Out Partitions on `/dev/did/dsk/d5`

```
# format /dev/did/rdsk/d5s2
# format> partition
[ output deleted ]
# partition> print
Current partition table (unnamed):
Total disk cylinders available: 34530 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       1 - 34529      269.77GB    (34529/0/0)  565723136
  1        usr    wm       0 - 0            0         (0/0/0)
  2     backup    wu       0 - 34529      269.77GB    (34530/0/0)  565739520
  3 unassigned    wu       0                0         (0/0/0)              0
  4 unassigned    wu       0                0         (0/0/0)              0
  5 unassigned    wu       0                0         (0/0/0)              0
  6 unassigned    wu       0                0         (0/0/0)              0
  7 unassigned    wu       0                0         (0/0/0)              0

NOTE: Partition 2 (backup) will not be used and was created by format(1M) by
default.
```

3. Replicate the device `d5` partitioning to devices `d6` through `d8`.

   This example shows the command for device `d6`.

```
#prtvtoc /dev/did/rdsk/d5s2 | fmthard -s - /dev/did/rdsk/
d6s2
```

4. On all nodes that are potential hosts of the file systems, perform the following:

a. Configure the six partitions into two StorageTek QFS shared file systems by adding two new configuration entries (`qfs1` and `qfs2`) to the `mcf` file.

Figure 124. Adding Configuration Entries to the `mcf` File

```
# cat >> /etc/opt/SUNWsamfs/mcf <<EOF
#
# StorageTek QFS file system configurations
#
# Equipment   Equipment     Equipment   Family    Device   Additional
# Identifier    Ordinal      Type         Set      State   Parameters
# ----------------- ---------    ---------    -------    ------  ----------
qfs1         100         ma          qfs1         -       shared
/dev/did/dsk/d4s0     101         mm          qfs1        -
/dev/did/dsk/d5s0     102         mr          qfs1        -
/dev/did/dsk/d6s0     103         mr          qfs1        -

qfs2         200         ma          qfs2         -       shared
/dev/did/dsk/d4s1     201         mm          qfs2        -
/dev/did/dsk/d7s0     202         mr          qfs2        -
/dev/did/dsk/d8s0     203         mr          qfs2        -

EOF
```

For more information about the `mcf` file, see the *StorageTek ASM Installation and Configuration Guide*.

b. Edit the `/etc/opt/SUNWsamfs/samfs.cmd` file to add the mount options that are required for the Sun Cluster data service for Oracle Real Application Clusters.

Figure 125. Example `samfs.cmd` File

```
fs = qfs2
   stripe = 1
   sync_meta = 1
   mh_write
   qwrite
   forcedirectio
   nstreams = 1024
   rdlease = 600
```

For more information about the mount options that are required by the Sun Cluster data service for Oracle Real Application Clusters, see the *Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS*.

c. Validate that the configuration is correct.

Be sure to perform this validation after you have configured the `mcf` file and the `samfs.cmd` file on each node.

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

To Create the StorageTek QFS Shared File System and Configure Sun Cluster Nodes

Perform this procedure for each file system you are creating. This example describes how to create the `qfs1` file system.

1. Obtain the Sun Cluster private interconnect names by using the following command.

Figure 126. Obtaining the Sun Cluster Private Interconnect Names

```
# /usr/cluster/bin/scconf -p |egrep "Cluster node name:|Node private \
hostname:"
Cluster node name:                                    scnode-A
  Node private hostname:                              clusternode1-priv
Cluster node name:                                    scnode-B
  Node private hostname:                              clusternode2-priv
```

2. On all nodes that are potential hosts of the file system, perform the following:

   a. Use the `samd`(1M) `config` command, which signals to the StorageTek QFS daemon that a new StorageTek QFS configuration is available.

```
# samd config
```

   b. Create the StorageTek QFS shared hosts file for the file system (`/etc/opt/SUNWsamfs/hosts.`*family-set-name*), based on the Sun Cluster's private interconnect names that you obtained in Step 1.

3. Edit the unique StorageTek QFS shared file system's host configuration file with the Sun Cluster interconnect names.

For Sun Cluster failover and fencing operations, the StorageTek QFS shared file system must use the same interconnect names as the Sun Cluster system.

Figure 127. Editing Each File System's Host Configuration File

```
# cat > hosts.qfs1 <<EOF
# File  /etc/opt/SUNWsamfs/hosts.qfs1
# Host          Host IP                                  Server   Not  Server
# Name          Addresses                                Priority Used Host
# ------------- ---------------------------------------- -------- ---- ----
scnode-A        clusternode1-priv                          1        -    server
scnode-B        clusternode2-priv                          2        -

EOF
```

4. From one node in the Sun Cluster, use the `sammkfs`(1M) `-S` command to create the StorageTek QFS shared file system.

```
# sammkfs -S qfs1 < /dev/null
```

5. On all nodes that are potential hosts of the file system, perform the following:

   a. Use the `mkdir`(1M) command to create a global mount point for the file system, use the `chmod`(1M) command to make `root` the owner of the mount point, and use the chown(1M) command to make the mount point usable by other with read/write (755) access.

Figure 128. Creating a Global Mount Point for the qfs1 File System

```
# mkdir /global/qfs1
# chmod 755 /global/qfs1
# chown root:other /global/qfs1
```

b. Add the StorageTek QFS shared file system entry to the /etc/vfstab file.

Figure 129. Adding the Shared File System Entry to the /etc/vfstab File

```
# cat >> /etc/vfstab <<EOF
# device       device       mount       FS      fsck    mount      mount
# to mount     to fsck      point       type    pass    at boot    options
#
qfs1              -     /global/qfs1    samfs    -        no        shared
EOF
```

To Validate the Configuration

Perform this procedure for each file system you create. This example describes how to validate the configuration for file system qfs1.

1. If you do not know which node is acting as the metadata server for the file system, use the samsharefs(1M) -R command.

Figure 130. Determining Which Node is the Metadata Server

```
# samsharefs -R qfs1
#
# Host file for family set 'qfs1'
#
# Version: 4    Generation: 1    Count: 2
# Server = host 1/scnode-A, length = 165
#
scnode-A clusternode2-priv 1 - server
scnode-B clusternode2-priv 2 -
```

The example shows that the metadata server for qfs1 is scnode-A.

2. Use the mount(1M) command to mount the file system first on the metadata server and then on each node in the Sun Cluster system.

It is very imporant that you mount the file system on the metadata server first.

Figure 131. Mounting File System, `qfs1`, on a Sun Cluster Node

```
# mount qfs1
# ls /global/qfs1
lost+found/
```

3. Validate voluntary failover by issuing the `samsharefs`(1M) `-s` command, which changes the StorageTek QFS shared file system between nodes.

Figure 132. Switching Over File System `qfs1` to Validate Voluntary Failover

```
# samsharefs -s scnode-B qfs1
# ls /global/qfs1
lost+found/
# samsharefs -s scnode-A qfs1
# ls /global/qfs1
lost+found
```

4. Validate that the required Sun Cluster resource type is added to the resource configuration.

```
# scrgadm -p | egrep “SUNW.qfs”
```

5. If you cannot find the Sun Cluster resource type, use the `scrgadm`(1M) `-a -t` command to add it to the resource configuration.

```
#scrgadm -a -t SUNW.qfs
```

6. Register and configure the `SUNW.qfs` resource type.

Figure 133. Configuring the `SUNW.qfs` Resource

```
# scrgadm -a -g qfs-rg -h scnode-A,scnode-B
# scrgadm -a -g qfs-rg -t SUNW.qfs -j qfs-res \
           -x QFSFileSystem=/global/qfs1,/global/qfs2
```

7. Use the `scswitch`(1M) `-Z -g` command to bring the resource group online.

```
#scswitch -Z -g qfs-rg
```

8. Ensure that the resource group is functional on all configured nodes.

Figure 134. Testing the Resource Group on Configured Nodes

```
# scswitch -z -g qfs-rg -h scnode-B
# scswitch -z -g qfs-rg -h scnode-A
```

To Configure the Sun Cluster Data Service for Oracle Real Application Clusters

This section provides an example of how to configure the data service for Oracle Real Application Clusters for use with StorageTek QFS shared file systems. For more information, see the *Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS*.

1. Install the data service as described in the *Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS*.
2. Mount the StorageTek QFS shared file systems.
3. Set the correct ownership and permissions on the file systems so that the Oracle database operations are successful.

Figure 135. Setting Ownership and Permissions on the File Systems `qfs1` **and** `qfs2`

```
# chown oracle:dba /global/qfs1 /global/qfs2
# chmod 755 /global/qfs1 /global/qfs2
```

4. As the `oracle` user, create the subdirectories that are required for the Oracle Real Application Clusters installation and database files.

Figure 136. Creating Subdirectories Within File Systems `qfs1` **and** `qfs2`

```
$ id
uid=120(oracle) gid=520(dba)
$ mkdir /global/qfs1/oracle_install
$ mkdir /global/qfs2/oracle_db
```

The Oracle Real Application Clusters installation uses the `/global/qfs1/oracle_install` directory path as the value for the ORACLE_HOME environment variable that is used in Oracle operations. The Oracle Real Application Clusters database files' path is prefixed with the `/global/qfs2/oracle_db` directory path.

5. Install the Oracle Real Application Clusters software.

During the installation, provide the path for the installation as defined in Step 4 (`/global/qfs1/oracle_install`).

6. Create the Oracle Real Application Clusters database.

   During database creation, specify that you want the database files located in the `qfs2` shared file system.

7. If you are automating the startup and shutdown of Oracle Real Application Clusters database instances, ensure that the required dependencies for resource groups and resources are set.

   For more information, see the *Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS*.

Note: If you plan to automate the startup and shutdown of Oracle Real Application Clusters database instances, you must use Sun Cluster 3.1 9/04 or a compatible version.

Configuring an Unshared File System on a Sun Cluster

When you install the unshared StorageTek QFS file system on a Sun Cluster system, you configure the file system for high availability (HA) under the Sun Cluster `HAStoragePlus` resource type. An unshared StorageTek QFS file system on a Sun Cluster is typically associated with one or more failover applications, such as HA-NFS, HA-ORACLE, and so on. Both the unshared StorageTek QFS file system and the failover applications are active in a single resource group; the resource group is active on one Sun Cluster node at a time.

An unshared StorageTek QFS file system is mounted on a single node at any given time. If the Sun Cluster fault monitor detects an error, or if you switch over the resource group, the unshared StorageTek QFS file system and its associated HA applications fail over to another node, depending on how the resource group has been previously configured.

Any file system contained on a Sun Cluster global device group (`/dev/global/*`) can be used with the `HAStoragePlus` resource type. When a file system is configured with the `HAStoragePlus` resource type, it becomes part of a Sun Cluster resource group and the file system under Sun Cluster Resource Group Manager (RGM) control is mounted locally on the node where the resource group is active. When the RGM causes a resource group switchover or fails over to another configured Sun Cluster node, the unshared StorageTek QFS file system is unmounted from the current node and remounted on the new node.

Each unshared StorageTek QFS file system requires a minimum of two raw disk partitions or volume manager-controlled volumes (Solstice DiskSuite/ Solaris Volume Manager or VERITAS Clustered Volume Manager), one for StorageTek QFS metadata (inodes) and one for StorageTek QFS file data. Configuring multiple partitions or volumes across multiple disks through multiple data paths increases unshared StorageTek QFS file system performance. For information about sizing metadata and file data partitions, see “Design Basics” on page 7.

This section provides three examples of Sun Cluster configurations using the unshared StorageTek QFS file system. In these examples, a file system is configured in combination with an HA-NFS file mount point on the following:

- Raw global devices in Example 1
- Solstice DiskSuite/Solaris Volume Manager-controlled volumes in Example 2
- VERITAS Clustered Volume Manager (VxVM)-controlled volumes in Example 3

For simplicity in all of these configurations, ten percent of each file system is used for StorageTek QFS metadata and the remaining space is used for StorageTek QFS file data. For information about sizing and disk layout considerations, see the *StorageTek ASM Installation and Configuration Guide*.

Example 1

This example shows how to configure the unshared StorageTek QFS file system with HA-NFS on raw global devices. For this configuration, the raw global devices must be contained on controller-based storage. This controller-based storage must support device redundancy by using RAID-1 or RAID-5.

As shown in Figure 121., the DID devices used in this example, `d4` through `d7`, are highly available and are contained on controller-based storage. (This example uses devices `d4` through `d7.)`The `HAStoragePlus` resource type requires the use of global devices, so each DID device (`/dev/did/dsk/dx`) is accessible as a global device by using the following syntax: `/dev/global/dsk/dx`.

The main steps in this example are as follows:

1. Prepare to create an unshared file system.
2. Create the file system and configure the Sun Cluster nodes.
3. Configure the network name service and the IPMP validation testing.
4. Configure HA-NFS and configure the file system for high availability.

To Prepare to Create an Unshared StorageTek QFS File System

1. Use the `format`(1M) utility to lay out the partitions on `/dev/global/dsk/d4`.

Figure 137. Command that Lays Out Partitions on `/dev/global/dsk/d4`.

```
# format /dev/global/rdsk/d4s2
# format> partition
[ output deleted ]
# partition> print
Current partition table (original):
Total disk cylinders available: 34530 + 2 (reserved cylinders)
Part      Tag     Flag      Cylinders         Size           Blocks
 0   unassigned    wm       1 -  3543        20.76GB    (3543/0/0)   43536384
 1   unassigned    wm    3544 - 34529       181.56GB    (30986/0/0) 380755968
 2   backup        wu       0 - 34529       202.32GB    (34530/0/0) 424304640
 3   unassigned    wu       0                 0         (0/0/0)             0
 4   unassigned    wu       0                 0         (0/0/0)             0
 5   unassigned    wu       0                 0         (0/0/0)             0
 6   unassigned    wu       0                 0         (0/0/0)             0
 7   unassigned    wu       0                 0         (0/0/0)             0

NOTE: Partition 2 (backup) will not be used and was created by format(1m) by
default.
```

Partition (or slice) 0 skips over the volume's Volume Table of Contents (VTOC) and is then configured as a 20 GB partition. The remaining space is configured into partition 1.

2. Replicate the global device `d4` partitioning to global devices `d5` through `d7`.

   This example shows the command for global device `d5`.

```
# prtvtoc /dev/global/rdsk/d4s2 | fmthard \
-s - /dev/global/rdsk/d5s2
```

3. On all nodes that are potential hosts of the file system, perform the following:

a. Configure the eight partitions (four global devices, with two partitions each) into a StorageTek QFS file system by adding a new file system entry to the `mcf` file.

Figure 138. Adding the New File System to the `mcf` File

```
# cat >> /etc/opt/SUNWsamfs/mcf <<EOF

#
# StorageTek QFS file system configurations
#
# Equipment          Equipment    Equipment  Family    Device  Additional
# Identifier         Ordinal       Type       Set      State   Parameters
# --------------     ---------    ---------  -------   ------  -----------
qfsnfs1              100           ma     qfsnfs1     on
/dev/global/dsk/d4s0   101           mm     qfsnfs1
/dev/global/dsk/d5s0   102           mm     qfsnfs1
/dev/global/dsk/d6s0   103           mm     qfsnfs1
/dev/global/dsk/d7s0   104           mm     qfsnfs1
/dev/global/dsk/d4s1   105           mr     qfsnfs1
/dev/global/dsk/d5s1   106           mr     qfsnfs1
/dev/global/dsk/d6s1   107           mr     qfsnfs1
/dev/global/dsk/d7s1   108           mr     qfsnfs1
EOF
```

For information about the `mcf` file, see the *StorageTek ASM Installation and Configuration Guide*.

b. Validate that the configuration information you added to the `mcf` file is correct.

It is important to complete this step before you configure the StorageTek QFS file system under the `HAStoragePlus` resource type.

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

Step 2: Create the StorageTek QFS File System and Configure The Sun Cluster Nodes

1. On all nodes that are potential hosts of the file system, use the `samd`(1M) `config` command, which signals to the StorageTek QFS daemon that a new StorageTek QFS configuration is available.

```
# samd config
```

2. From one node in the Sun Cluster, use the `sammkfs`(1M) command to create the file system.

```
# sammkfs qfsnfs1 < /dev/null
```

3. On all nodes that are potential hosts of the file system, perform the following:

   a. Use the `mkdir`(1M) command to create a global mount point for the file system, use the `chmod`(1M) command to make `root` the owner of the mount point, and use the `chown`(1M) command to make the mount point usable by `other` with read/write (755) access.

Figure 139. Creating a Global Mount Point for File System `qfsnfs1`

```
# mkdir /global/qfsnfs1
# chmod 755 /global/qfsnfs1
# chown root:other /global/qfsnfs1
```

   b. Add the StorageTek QFS file system entry to the `/etc/vfstab` file.

   Note that the mount options field contains the `sync_meta=1` value.

Figure 140. Adding the File System Entry to the `/etc/vfstab` **File**

```
# cat >> /etc/vfstab <<EOF

# device       device         mount        FS       fsck     mount      mount
# to mount     to fsck        point        type     pass     at boot    options
#
qfsnfs1          -       /global/qfsnfs1    samfs       2         no
sync_meta=1

EOF
```

c. Validate the configuration by mounting and unmounting the file system.

Figure 141. Validating the Configuration

```
# mount qfsnfs1
# ls /global/qfsnfs1
lost+found/
# umount qfsnfs1
```

4. Use the `scrgadm`(1M) -`p | egrep` command to validate that the required Sun Cluster resource types have been added to the resource configuration.

Figure 142. Searching for the Required Sun Cluster Resource Types

```
# scrgadm -p | egrep
"SUNW.HAStoragePlus|SUNW.LogicalHostname|SUNW.nfs"
```

5. If you cannot find a required Sun Cluster resource type, use the `scrgadm`(1M) `-a -t` command to add it to the configuration.

Figure 143. Adding the Required Sun Cluster Resource Types

```
# scrgadm -a -t SUNW.HAStoragePlus
# scrgadm -a -t SUNW.LogicalHostname
# scrgadm -a -t SUNW.nfs
```

To Configure the Network Name Service and the IPMP Validation Testing

This section provides an example of how to configure the network name service and the IPMP Validation Testing for your Sun Cluster nodes. For more information, see the *Sun Cluster Software Installation Guide for Solaris OS*.

1. Use `vi` or another text editor to edit the `/etc/nsswitch.conf` file so that it looks in the Sun Cluster and files for node names.

   Perform this step before you configure the NIS server.

Figure 144. Editing the `/etc/nsssswitch` File to Look in the Sun Cluster and Files for Node Names

```
# cat /etc/nsswitch.conf
#
# /etc/nsswitch.nis:
#
# An example file that could be copied over to /etc/nsswitch.conf; it
# uses NIS (YP) in conjunction with files.
#
# the following two lines obviate the "+" entry in /etc/passwd and /
etc/group.
passwd:    files nis
group:     files nis

# Cluster s/w and local /etc/hosts file take precedence over NIS
hosts:    cluster files nis [NOTFOUND=return]
ipnodes:  files
# Uncomment the following line and comment out the above to resolve
# both IPv4 and IPv6 addresses from the ipnodes databases. Note that
# IPv4 addresses are searched in all of the ipnodes databases before
# searching the hosts databases. Before turning this option on, consult
# the Network Administration Guide for more details on using IPv6.
# ipnodes: nis [NOTFOUND=return] files

networks: nis[NOTFOUND=return] files
protocols: nis [NOTFOUND=return] files
rpc: nis[NOTFOUND=return] files
ethers: nis[NOTFOUND=return] files
netmaks: nis[NOTFOUND=return] files
bootparams: nis[NOTFOUND=return] files
publickey: nis[NOTFOUND=return] files

netgroup: nis

automount: files nis
aliases: files nis
[remainder of file content not shown]
```

2. Verify that the changes you made to the `/etc/nsswitch.conf` are correct.

Figure 145. Verifying the `/etc/nsswitch.conf` File Changes

```
# grep '^hosts:' /etc/nsswitch.conf
hosts:    cluster files nis [NOTFOUND=return]
#
```

3. Set up IPMP validation testing by using available network adapters.

   The adapters `qfe2` and `qfe3` are used as examples.

   a. Statically configure the IPMP test address for each adapter.

Figure 146. Statically Configuring the IPMP Test Address for Adapters `qfe2` and `qfe3`

```
#cat >> /etc/hosts << EOF

#
# Test addresses for scnode-A
#
192.168.2.2      `uname -n`-qfe2
192.168.2.3      `uname -n`-qfe2-test
192.168.3.2      `uname -n`-qfe3
192.168.3.3      `uname -n`-qfe3-test

#
# Test addresses for scnode-B
#
192.168.2.4      `uname -n`-qfe2
192.168.2.5      `uname -n`-qfe2-test
192.168.3.4      `uname -n`-qfe3
192.168.3.5      `uname -n`-qfe3-test
EOF
```

b. Dynamically configure the IPMP Adapters

Figure 147. Dynamically Configuring the IPMP Adapters, `qfe2` **and** `qfe3`

```
# ifconfig qfe2 plumb `uname -n`-qfe2-test netmask + broadcast + deprecated \
        -failover -standby group ipmp0 up
# ifconfig qfe2 addif `uname -n`-qfe2 up
# ifconfig qfe3 plumb `uname -n`-qfe3-test netmask + broadcast + deprecated \
        -failover -standby group ipmp0 up
# ifconfig qfe3 addif `uname -n`-qfe3 up
```

c. Verify the configuration.

Figure 148. Verifying the Configuration of the IPMP Adapters, `qfe2` **and** `qfe3`

```
# cat > /etc/hostname.qfe2 << EOF
`uname -n`-qfe2-test netmask + broadcast + deprecated -failover -standby \
        group ipmp0 up addif `uname -n`-qfe2 up
EOF

# cat > /etc/hostname.qfe3 << EOF
`uname -n`-qfe3-test netmask + broadcast + deprecated -failover -standby \
        group ipmp0 up addif `uname -n`-qfe3 up
EOF
```

To Configure HA-NFS and the StorageTek QFS File System for High Availability

This section provides an example of how to configure HA-NFS. For more information about HA-NFS, see the *Sun Cluster Data Service for Network File System (NFS) Guide for Solaris OS* and your NFS documentation.

1. Create the NFS share point for the StorageTek QFS file system.

   Note that the share point is contained within the `/global` file system, not within the StorageTek QFS file system.

Figure 149. Creating the NFS Share Points for the Two File Systems

```
# mkdir -p /global/nfs/SUNW.nfs
# echo "share -F nfs -o rw /global/qfsnfs1" > \ /global/nfs/SUNW.nfs/
dfstab.nfs1-res
```

2. Create the NFS resource group.

```
#scrgadm -a -g nfs-rg -y PathPrefix=/global/nfs
```

3. Add the NFS logical host to the `/etc/hosts` table, using the address for your site.

Figure 150. Adding the NFS Logical Host to the `/etc/hosts` Table

```
# cat >> /etc/hosts << EOF
#
# IP Addresses for LogicalHostnames
#
192.168.2.10     lh-qfs1
EOF
```

4. Use the `scrgadm`(1M) `-a -L -g` command to add the logical host to the NFS resource group.

```
#scrgadm -a -L -g nfs-rg -l lh-nfs1
```

5. Use the `scrgadm`(1M) `-c -g` command to configure the `HAStoragePlus` resource type.

Figure 151. Configuring the `HAStoragePlus` Resource Type

```
# scrgadm -c -g nfs-rg -h scnode-A,scnode-B
# scrgadm -a -g nfs-rg -j qfsnfs1-res -t SUNW.HAStoragePlus \
        -x FilesystemMountPoints=/global/qfsnfs1 \
        -x FilesystemCheckCommand=/bin/true
```

6. Bring the resource group online.

```
#scswitch -Z -g nfs-rg
```

7. Configure the NFS resource type and set a dependency on the `HAStoragePlus` resource.

Figure 152. Configuring the NFS Resource Type to Depend on the `HAStoragePlus` Resource

```
# scrgadm -a -g nfs-rg -j nfs1-res -t SUNW.nfs -y \
Resource_dependencies=qfsnfs1-res
```

8. Bring the NFS resource online.

```
# scswitch -e -j nfs1-res
```

The NFS resource `/net/lh-nfs1/global/qfsnfs1` is now fully configured and is also highly available.

9. Before announcing the availability of the highly available NFS file system on the StorageTek QFS file system, ensure that the resource group can be switched between all configured nodes without errors and can be taken online and offline.

Figure 153. Testing the Resource Groups

```
# scswitch -z -g nfs-rg -h scnode-A
# scswitch -z -g nfs-rg -h scnode-B

# scswitch -F -g nfs-rg

# scswitch -Z -g nfs-rg
```

Example 2

This example shows how to configure the unshared StorageTek QFS file system with HA-NFS on volumes controlled by Solstice DiskSuite/Solaris Volume Manager software. With this configuration, you can choose whether the DID devices are contained on redundant controller-based storage using RAID-1 or RAID-5 volumes. Typically, Solaris Volume Manager is used only when the underlying controller-based storage is not redundant.

As shown in Figure 121., the DID devices used in this example, `d4` through `d7`, are highly available and are contained on controller-based storage. Solaris Volume Manager requires that DID devices be used to populate the raw devices from which Solaris Volume Manager can configure volumes. Solaris Volume Manager creates globally accessible disk groups, which can then be used by the `HAStoragePlus` resource type for creating StorageTek QFS file systems.

This example follows these steps:

1. Prepare the Solstice DiskSuite/Solaris Volume Manager software.
2. Prepare to create an unshared file system.
3. Create the file system and configure the Sun Cluster nodes.
4. Configure the network name service and the IPMP validation testing.
5. Configure HA-NFS and configure the file system for high availability.

To Prepare the Solstice DiskSuite/Solaris Volume Manager Software

1. Determine whether a Solaris Volume Manager metadatabase (`metadb`) is already configured on each node that is a potential host of the StorageTek QFS file system.

Figure 154. Determining Whether a Solaris Volume Manager Metadatabase is Already Configured

```
# metadb
        flags           first blk       block count
     a m  p  luo        16              8192            /dev/dsk/c0t0d0s7
     a    p  luo        16              8192            /dev/dsk/c1t0d0s7
     a    p  luo        16              8192            /dev/dsk/c2t0d0s7
```

If the `metadb`(1M) command does not return a metadatabase configuration, then on each node, create three or more database replicas on one or more local disks. Each replica must be at least 16 MB in size. For more information about creating the metadatabase configuration, see the *Sun Cluster Software Installation Guide for Solaris OS*.

2. Create an HA-NFS disk group to containall Solaris Volume Manager volumes for this StorageTek QFS file system.

```
# metaset -s nfsdg -a -h scnode-A scnode-B
```

3. Add DID devices `d4` through `d7` to the pool of raw devices from which Solaris Volume Manager can create volumes.

Figure 155. Adding DID Devices `d4` Through `d7` to the Pool of Raw Devices

```
# metaset -s nfsdg -a /dev/did/dsk/d4 /dev/did/dsk/d5 \
        /dev/did/dsk/d6 /dev/did/dsk/d7
```

To Prepare to Create a StorageTek QFS File System

1. Use the `format`(1M) utility to lay out partitions on `/dev/global/dsk/d4`.

Figure 156. Command that Lays Out Partitions on `/dev/global/dsk/d4`**.**

```
# format /dev/global/rdsk/d4s2
# format> partition
[ output deleted ]
# partition> print
Current partition table (original):
Total disk cylinders available: 34530 + 2 (reserved cylinders)
Part      Tag      Flag      Cylinders         Size           Blocks
 0   unassigned     wm        1 -  3543        20.76GB    (3543/0/0)   43536384
 1   unassigned     wm     3544 - 34529       181.56GB    (30986/0/0) 380755968
 2   backup         wu        0 - 34529       202.32GB    (34530/0/0) 424304640
 3   unassigned     wu        0                 0         (0/0/0)             0
 4   unassigned     wu        0                 0         (0/0/0)             0
 5   unassigned     wu        0                 0         (0/0/0)             0
 6   unassigned     wu        0                 0         (0/0/0)             0
 7   unassigned     wu        0                 0         (0/0/0)             0

NOTE: Partition 2 (backup) will not be used and was created by format(1m) by
default.
```

Figure 156. shows that partition or slice 0 skips over the volume's Volume Table of Contents (VTOC) and is then configured as a 20 GB partition. The remaining space is configured into partition 1.

2. Replicate the partitioning of DID device `d4` to DID devices `d5` through `d7`.

   This example shows the command for device `d5`.

```
#prtvtoc /dev/global/rdsk/d4s2 | fmthard \
-s - /dev/global/rdsk/d5s2
```

3. Configure the eight partitions (four DID devices, two partitions each) into two RAID-1 (mirrored) StorageTek QFS metadata volumes and two RAID-5 (parity-striped) StorageTek QFS file data volumes.

Combine partition (slice) 0 of these four drives into two RAID-1 sets.

Figure 157. Configuring Partitions Into RAID-1 Metadata and Into RAID-5 Data Volumes, and Then Combining Partition Zero of the Four Drives into RAID-1 Sets

```
# metainit -s nfsdg -f d1 1 1 /dev/did/dsk/d4s0
# metainit -s nfsdg -f d2 1 1  /dev/did/dsk/d5s0
# metainit -s nfsdg d10 -m d1 d2
# metainit -s nfsdg -f d3 1 1 /dev/did/dsk/d6s0
# metainit -s nfsdg -f d4 1 1  /dev/did/dsk/d7s0
# metainit -s nfsdg d11 -m d3 d4
```

4. Combine partition 1 of these four drives into two RAID-5 sets.

Figure 158. Combining Partition One of the Four Drives Into Two RAID-5 Sets

```
# metainit -s nfsdg d20 -p /dev/did/dsk/d4s1 205848574b
# metainit -s nfsdg d21 -p /dev/did/dsk/d5s1 205848574b
# metainit -s nfsdg d22 -p /dev/did/dsk/d6s1 205848574b
# metainit -s nfsdg d23 -p /dev/did/dsk/d7s1 205848574b
# metainit -s nfsdg d30 -r d20 d21 d22 d23
```

5. On each node that is a potential host of the file system, add the StorageTek QFS file system entry to the `mcf` file.

Figure 159. Adding the StorageTek QFS File System to the Metadata Server's `mcf` File

```
# cat >> /etc/opt/SUNWsamfs/mcf <<EOF

# StorageTek QFS file system configurations
#
# Equipment           Equipment   Equipment   Family   Device    Additional
# Identifier            Ordinal     Type         Set   State     Parameters
# ------------------ ---------    ---------   -------   ------    ----------
qfsnfs1                 100          ma      qfsnfs1     on
/dev/md/nfsdg/dsk/d10    101          mm      qfsnfs1
/dev/md/nfsdg/dsk/d11    102          mm      qfsnfs1
/dev/md/nfsdg/dsk/d30    103          mr      qfsnfs1
EOF
```

For more information about the `mcf` file, see the *StorageTek ASM Installation and Configuration Guide*.

6. Validate that the `mcf` configuration is correct on each node.

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

To Create the StorageTek QFS File System and Configure Sun Cluster Nodes

1. On each node that is a potential host of the file system, use the `samd`(1M) `config` command.

   This command signals to the StorageTek QFS daemon that a new StorageTek QFS configuration is available.

```
# samd config
```

2. Enable Solaris Volume Manager mediation detection of disk groups, which assists the Sun Cluster system in the detection of drive errors.

Figure 160. Enabling Solaris Volume Manager Mediation Detection of Disk Groups

```
# metaset -s nfsdg -a -m scnode-A
# metaset -s nfsdg -a -m scnode-B
```

3. On each node that is a potential host of the file system, ensure that the NFS disk group exists.

```
# metaset -s nfsdg -t
```

4. From one node in the Sun Cluster system, use the `sammkfs`(1M) command to create the StorageTek QFS file system.

```
# sammkfs qfsnfs1 < /dev/null
```

5. On each node that is a potential host of the file system, perform the following:

   a. Use the `mkdir`(1M) command to create a global mount point for the file system, use the `chmod`(1M) command to make `root` the owner of the mount point, and use the `chown`(1M) command to make the mount point usable by `other` with read/write (755) access.

Figure 161. Creation of a Global Mount Point for the `qfsnfs1` File System

```
# mkdir /global/qfsnfs1
# chmod 755 /global/qfsnfs1
# chown root:other /global/qfsnfs1
```

b. Add the StorageTek QFS file system entry to the `/etc/vfstab` file.

   Note that the mount options field contains the `sync_meta=1` value.

Figure 162. Editing the `/etc/vfstab` File to Add the File System Entry

```
# cat >> /etc/vfstab << EOF
# device        device        mount          FS      fsck    mount      mount
# to mount      to fsck       point          type    pass    at boot    options
#
qfsnfs1          -     /global/qfsnfs1   samfs    2       no      sync_meta=1
EOF
```

c. Ensure that the nodes are configured correctly by mounting and unmounting the file system.

   Perform this step one node at a time. In this example, the `qfsnfs1` file system is being mounted and unmounted on one node.

Figure 163. Validating the Configuration

```
# mount qfsnfs1
# ls /global/qfsnfs1
lost+found/
# umount qfsnfs1
```

Note: When testing the mount point, use the `metaset -r` (release) and `-t` (take) command to move the `nfsdg` disk group between Sun Cluster nodes. Then use the `samd`(1M) `config` command to alert the daemon of the configuration changes.

6. Use the `scrgadm`(1M) `-p | egrep` command to validate that the required Sun Cluster resource types have been added to the resource configuration.

```
# scrgadm -p | egrep
“SUNW.HAStoragePlus|SUNW.LogicalHostname|SUNW.nfs”
```

If you cannot find a required Sun Cluster resource type, add it with one or more of the following commands.

Figure 164. Adding the Resource Types to the Resource Configuration

```
# scrgadm -a -t SUNW.HAStoragePlus
# scrgadm -a -t SUNW.LogicalHostname
# scrgadm -a -t SUNW.nfs
```

To Configure the Network Name Service and the IPMP Validation Testing

This section provides an example of how to configure the network name service and IPMP validation testing for use with the StorageTek QFS software. For more information, see the *System Administration Guide: IP Services* and the *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

1. Use `vi` or another text editor to edit the `/etc/nsswitch.conf` file so that it looks in the Sun Cluster and files for node names.

   Perform this step before you configure the NIS server.

Figure 165. Editing the `/etc/nsssswitch` **File to Look in the Sun Cluster and Files for Node Names**

```
# cat /etc/nsswitch.conf
#
# /etc/nsswitch.nis:
#
# An example file that could be copied over to /etc/nsswitch.conf; it
# uses NIS (YP) in conjunction with files.
#
# the following two lines obviate the "+" entry in /etc/passwd and /
etc/group.
passwd:    files nis
group:     files nis

# Cluster s/w and local /etc/hosts file take precedence over NIS
hosts:    cluster files nis [NOTFOUND=return]
ipnodes:  files
# Uncomment the following line and comment out the above to resolve
# both IPv4 and IPv6 addresses from the ipnodes databases. Note that
# IPv4 addresses are searched in all of the ipnodes databases before
# searching the hosts databases. Before turning this option on, consult
# the Network Administration Guide for more details on using IPv6.
# ipnodes: nis [NOTFOUND=return] files

networks: nis[NOTFOUND=return] files
protocols: nis [NOTFOUND=return] files
rpc: nis[NOTFOUND=return] files
ethers: nis[NOTFOUND=return] files
netmaks: nis[NOTFOUND=return] files
bootparams: nis[NOTFOUND=return] files
publickey: nis[NOTFOUND=return] files

netgroup: nis

automount: files nis
aliases: files nis
[remainder of file content not shown]
```

2. Verify that the changes you made to the `/etc/nsswitch.conf` are correct.

Figure 166. Verifying the `/etc/nsswitch.conf` File Changes

```
# grep '^hosts:' /etc/nsswitch.conf
hosts:    cluster files nis [NOTFOUND=return]
#
```

3. Set up IPMP validation testing using available network adapters.

   The adapters `qfe2` and `qfe3` are used in the examples.

a. Statically configure the IPMP test address for each adapter.

Figure 167. Statically Configuring the IPMP Test Address for Each Adapter

```
# cat >> /etc/hosts << EOF
#
# Test addresses for scnode-A
#
192.168.2.2      `uname -n`-qfe2
192.168.2.3      `uname -n`-qfe2-test
192.168.3.2      `uname -n`-qfe3
192.168.3.3      `uname -n`-qfe3-test
#
# Test addresses for scnode-B
#
192.168.2.4      `uname -n`-qfe2
192.168.2.5      `uname -n`-qfe2-test
192.168.3.4      `uname -n`-qfe3
192.168.3.5      `uname -n`-qfe3-test
#
# IP Addresses for LogicalHostnames
#
192.168.2.10     lh-qfs1

EOF
```

b. Dynamically configure the IPMP adapters.

Figure 168. Dynamically Configuring the IPMP Adapters

```
# ifconfig qfe2 plumb `uname -n`-qfe2-test netmask + broadcast + deprecated \
        -failover -standby group ipmp0 up
# ifconfig qfe2 addif `uname -n`-qfe2 up
# ifconfig qfe3 plumb `uname -n`-qfe3-test netmask + broadcast + deprecated \
        -failover -standby group ipmp0 up
# ifconfig qfe3 addif `uname -n`-qfe3 up
```

c. Validate the configuration.

Figure 169. Dynamically Configuring the IPMP Adapters

```
# cat > /etc/hostname.qfe2 << EOF
`uname -n`-qfe2-test netmask + broadcast + deprecated -failover -standby \
        group ipmp0 up addif `uname -n`-qfe2 up
EOF
# cat > /etc/hostname.qfe3 << EOF
`uname -n`-qfe3-test netmask + broadcast + deprecated -failover -standby \
        group ipmp0 up addif `uname -n`-qfe3 up
EOF
```

To Configure HA-NFS and the StorageTek QFS File System for High Availability

This section provides an example of how to configure HA-NFS. For more information about HA-NFS, see the *Sun Cluster Data Service for Network File System (NFS) Guide for Solaris OS* and your NFS documentation.

1. Create the NFS share point for the StorageTek QFS file system.

   Note that the share point is contained within the `/global` file system, not within the StorageTek QFS file system.

Figure 170. Creating the NFS Share Points for the File Systems

```
# mkdir -p /global/nfs/SUNW.nfs
# echo "share -F nfs -o rw /global/qfsnfs1" > \ /global/nfs/SUNW.nfs/
dfstab.nfs1-res
```

2. Create the NFS resource group.

```
# scrgadm -a -g nfs-rg -y PathPrefix=/global/nfs
```

3. Add a logical host to the NFS resource group.

```
# scrgadm -a -L -g nfs-rg -l lh-nfs1
```

4. Configure the `HAStoragePlus` resource type.

Figure 171. Configuring the `HAStoragePlus` Resource Type

```
# scrgadm -c -g nfs-rg -h scnode-A,scnode-B
# scrgadm -a -g nfs-rg -j qfsnfs1-res -t SUNW.HAStoragePlus \
        -x FilesystemMountPoints=/global/qfsnfs1 \
        -x FilesystemCheckCommand=/bin/true
```

5. Bring the resource group online.

```
# scswitch -Z -g nfs-rg
```

6. Configure the NFS resource type and set a dependency on the `HAStoragePlus` resource.

Figure 172. Configuring the NFS Resource Type

```
# scrgadm -a -g nfs-rg -j nfs1-res -t SUNW.nfs -y \
Resource_dependencies=qfsnfs1-res
```

7. Use the `scswitch`(1M) `-e -j` command to bring the NFS resource online.

```
# scswitch -e -j nfs1-res
```

   The NFS resource `/net/lh-nfs1/global/qfsnfs1` is fully configured and highly available.

8. Before you announce the availability of the highly available NFS file system on the StorageTek QFS file system, ensure that the resource group can be switched between all configured nodes without errors and can be taken online and offline.

Figure 173. Testing the Resource Group

```
# scswitch -z -g nfs-rg -h scnode-A
# scswitch -z -g nfs-rg -h scnode-B

# scswitch -F -g nfs-rg

# scswitch -Z -g nfs-rg
```

Example 3

This example shows how to configure the unshared StorageTek QFS file system with HA-NFS on VERITAS Clustered Volume manager-controlled volumes (VxVM volumes). With this configuration, you can choose whether the DID devices are contained on redundant controller-based storage using RAID-1 or RAID-5. Typically, VxVM is used only when the underlying storage is not redundant.

As shown in Figure 121., the DID devices used in this example, `d4` through `d7`, are highly available and are contained on controller-based storage. VxVM requires that shared DID devices be used to populate the raw devices from which VxVM configures volumes. VxVM creates highly available disk groups by registering the disk groups as Sun Cluster device groups. These disk groups are not globally accessible, but can be failed over, making them accessible to at least one node. The disk groups can be used by the `HAStoragePlus` resource type.

Note: The VxVM packages are separate, additional packages that must be installed, patched, and licensed. For information about installing VxVM, see the VxVM Volume Manager documentation.

To use StorageTek QFS software with VxVM, you must install the following VxVM packages:

- VRTSvlic
- VRTSvmdoc
- VRTSvmman
- VRTSvmpro
- VRTSvxvm
- VRTSob and VRTSobgui (optional GUI packages)

This example follows these steps:

1. Configure the VxVM software.
2. Prepare to create an unshared file system.
3. Create the file system and configure the Sun Cluster nodes.
4. Validate the configuration.
5. Configure the network name service and the IPMP validation testing.
6. Configure HA-NFS and configure the file system for high availability.

To Configure the VxVM Software

This section provides an example of how to configure the VxVM software for use with the StorageTek QFS software. For more detailed information about the VxVM software, see the VxVM documentation.

1. Determine the status of DMP (dynamic multipathing) for VERITAS.

```
# vxdmpadm listctlr all
```

2. Use the `scdidadm`(1M) utility to determine the HBA controller number of the physical devices to be used by VxVM.

As shown in the following example, the multi-node accessible storage is available from `scnode-A` using HBA controller `c6`, and from node `scnode-B using` controller `c7`.

Figure 174. Determining the HBA Controller Number of the Physical Devices

```
# scdidadm -L
[ some output deleted]
4   scnode-A:/dev/dsk/c6t60020F20000037D13E26595500062F06d0 /dev/did/dsk/d4
4   scnode-B:/dev/dsk/c7t60020F20000037D13E26595500062F06d0 /dev/did/dsk/d4
```

3. Use VxVM to configure all available storage as seen through controller `c6`.

```
# vxdmpadm getsubpaths ctlr=c6
```

4. Place all of this controller's devices under VxVM control.

```
# vxdiskadd fabric_
```

5. Create a disk group, create volumes, and then start the new disk group.

```
# /usr/sbin/vxdg init qfs-dg qfs-dg00=disk0 \
qfsdg01=disk1 qfsdg02=disk2 qfsdg03=disk3
```

Ensure that the previously started disk group is active on this system.

Figure 175. Validating that the Disk Group is Active on This System

```
# vxdg import nfsdg
# vxdg free
```

6. Configure two mirrored volumes for StorageTek QFS metadata and two volumes for StorageTek QFS file data volumes.

These mirroring operations are performed as background processes, given the length of time they take to complete.

Figure 176. Configure Metadata and Data Volumes

```
# vxassist -g nfsdg make m1 10607001b
# vxassist -g nfsdg mirror m1&
# vxassist -g nfsdg make m2 10607001b
# vxassist -g nfsdg mirror m2&
# vxassist -g nfsdg make m10 201529000b
# vxassist -g nfsdg mirror m10&
# vxassist -g nfsdg make m11 201529000b
# vxassist -g nfsdg mirror m11&
```

7. Configure the previously created VxVM disk group as a Sun Cluster-controlled disk group.

```
#scconf -a -D type=vxvm,name=nfsdg,nodelist=scnode-
A:scnode-B
```

To Prepare to Create a StorageTek QFS File System

Perform this procedure on each node that is a potential host of the file system.

1. Add the StorageTek QFS file system entry to the `mcf` file.

Figure 177. Addition of the File System to the `mcf` File

```
# cat >> /etc/opt/SUNWsamfs/mcf   <<EOF
# StorageTek QFS file system configurations
#
# Equipment             Equipment  Equipment  Family     Device    Additional
# Identifier            Ordinal    Type       Set        State     Parameters
# ------------------    --------    ---------  -------    ------    ----------
qfsnfs1                   100        ma        qfsnfs1     on
/dev/vx/dsk/nfsdg/m1      101        mm        qfsnfs1
/dev/vx/dsk/nfsdg/m2      102        mm        qfsnfs1
/dev/vx/dsk/nfsdg/m10     103        mr        qfsnfs1
/dev/vx/dsk/nfsdg/m11     104        mr        qfsnds1
EOF
```

For more information about the `mcf` file, see the *StorageTek ASM Installation and Configuration Guide*.

2. Validate that the mcf configuration is correct.

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

To Create the StorageTek QFS File System and Configure Sun Cluster Nodes

1. On each node that is a potential host of the file system, use the samd(1M) config command.

   This command signals to the StorageTek QFS daemon that a new StorageTek QFS configuration is available.

```
# samd config
```

2. From one node in the Sun Cluster system, use the sammkfs(1M) command to create the StorageTek QFS file system.

```
# sammkfs qfsnfs1 < /dev/null
```

3. On each node that is a potential host of the file system, perform the following:

   a. Use the mkdir(1M) command to create a global mount point for the file system, use the chmod(1M) command to make root the owner of the mount point, and use the chown(1M) command to make the mount point usable by other with read/write (755) access.

Figure 178. Creating a Global Mount Point for the qfsnfs1 File System

```
# mkdir /global/qfsnfs1
# chmod 755 /global/qfsnfs1
# chown root:other /global/qfsnfs1
```

   b. Add the StorageTek QFS file system entry to the /etc/vfstab file.

   Note that the mount options field contains the sync_meta=1 value.

Figure 179. Adding the File System Entry to the /etc/vfstab File

```
# cat >> /etc/vfstab << EOF
# device        device        mount         FS       fsck     mount       mount
# to mount      to fsck       point         type     pass     at boot     options
#
qfsnfs1            -     /global/qfsnfs1  samfs    2        no      sync_meta=1
EOF
```

To Validate the Configuration

1. Validate that all nodes that are potential hosts of the file system are configured correctly.

   To do this, move the disk group that you created in "To Configure the VxVM Software" on page 264 to the node, and mount and then unmount the file system. Perform this validation one node at a time.

Figure 180. Validating the Configuration

```
# scswitch -z -D nfsdg -h scnode-B
# mount qfsnfs1
# ls /global/qfsnfs1
lost+found/
# umount qfsnfs1
```

2. Ensure that the required Sun Cluster resource types have been added to the resource configuration. If you cannot find a required Sun Cluster

```
# scrgadm -p | egrep
"SUNW.HAStoragePlus|SUNW.LogicalHostname|SUNW.nfs"
```

   resource type, add it with one or more of the following commands.

Figure 181. Adding Sun Cluster Resources to the Resource Configuration

```
# scrgadm -a -t SUNW.HAStoragePlus
# scrgadm -a -t SUNW.LogicalHostname
# scrgadm -a -t SUNW.nfs
```

To Configure the Network Name Service and the IPMP Validation Testing

This section provides an example of how to configure the network name service and the IPMP validation testing. For more information, see the *Sun Cluster Software Installation Guide for Solaris OS*.

1. Use `vi` or another text editor to edit the `/etc/nsswitch.conf` file so that it looks in the Sun Cluster and files for node names.

   Perform this step before you configure the NIS server.

Figure 182. Editing the `/etc/nsssswitch` File to Look in the Sun Cluster and Files for Node Names

```
# cat /etc/nsswitch.conf
#
# /etc/nsswitch.nis:
#
# An example file that could be copied over to /etc/nsswitch.conf; it
# uses NIS (YP) in conjunction with files.
#
# the following two lines obviate the "+" entry in /etc/passwd and /
etc/group.
passwd:    files nis
group:     files nis

# Cluster s/w and local /etc/hosts file take precedence over NIS
hosts:    cluster files nis [NOTFOUND=return]
ipnodes:  files
# Uncomment the following line and comment out the above to resolve
# both IPv4 and IPv6 addresses from the ipnodes databases. Note that
# IPv4 addresses are searched in all of the ipnodes databases before
# searching the hosts databases. Before turning this option on, consult
# the Network Administration Guide for more details on using IPv6.
# ipnodes: nis [NOTFOUND=return] files

networks: nis[NOTFOUND=return] files
protocols: nis [NOTFOUND=return] files
rpc: nis[NOTFOUND=return] files
ethers: nis[NOTFOUND=return] files
netmaks: nis[NOTFOUND=return] files
bootparams: nis[NOTFOUND=return] files
publickey: nis[NOTFOUND=return] files

netgroup: nis

automount: files nis
aliases: files nis
[remainder of file content not shown]
```

2. Verify that the changes you made to the `/etc/nsswitch.conf` are correct.

Figure 183. Verifying the `/etc/nsswitch.conf` File Changes

```
# grep '^hosts:' /etc/nsswitch.conf
hosts:    cluster files nis [NOTFOUND=return]
#
```

3. Set up IPMP validation testing using available network adapters.

   The adapters `qfe2` and `qfe3` are used as examples.

a. Statically configure IPMP test address for each adapter.

Figure 184. Statically Configuring the IPMP Test Address for Each Adapter

```
# cat >> /etc/hosts << EOF
#
# Test addresses for scnode-A
#
192.168.2.2      `uname -n`-qfe2
192.168.2.3      `uname -n`-qfe2-test
192.168.3.2      `uname -n`-qfe3
192.168.3.3      `uname -n`-qfe3-test
#
# Test addresses for scnode-B
#
192.168.2.4      `uname -n`-qfe2
192.168.2.5      `uname -n`-qfe2-test
192.168.3.4      `uname -n`-qfe3
192.168.3.5      `uname -n`-qfe3-test
#
# IP Addresses for LogicalHostnames
#
192.168.2.10     lh-qfs1
EOF
```

b. Dynamically configure IPMP adapters.

Figure 185. Dynamically Configuring the IPMP Adapters

```
# ifconfig qfe2 plumb `uname -n`-qfe2-test netmask + broadcast + deprecated \
        -failover -standby group ipmp0 up
# ifconfig qfe2 addif `uname -n`-qfe2 up
# ifconfig qfe3 plumb `uname -n`-qfe3-test netmask + broadcast + deprecated \
        -failover -standby group ipmp0 up
# ifconfig qfe3 addif `uname -n`-qfe3 up
```

c. Validate the configuration.

Figure 186. Dynamically Configuring the IPMP Adapters

```
# cat > /etc/hostname.qfe2 << EOF
`uname -n`-qfe2-test netmask + broadcast + deprecated -failover -standby \
        group ipmp0 up addif `uname -n`-qfe2 up
EOF

# cat > /etc/hostname.qfe3 << EOF
`uname -n`-qfe3-test netmask + broadcast + deprecated -failover -standby \
        group ipmp0 up addif `uname -n`-qfe3 up
EOF
```

To Configure HA-NFS and the StorageTek QFS File System for High Availability

This section provides an example of how to configure HA-NFS. For more information about HA-NFS, see the *Sun Cluster Data Service for Network File System (NFS) Guide for Solaris OS* and your NFS documentation.

1. On each node that is a potential host of the file system, create the NFS share point for the StorageTek QFS file system.

   Note that the share point is contained within the `/global` file system, not within the StorageTek QFS file system.

Figure 187. Creating the NFS Share Point for the File System

```
# mkdir -p /global/qfsnfs1/SUNW.nfs
# echo "share -F nfs -o rw /global/qfsnfs1" > \ /global/
qfsnfs1/SUNW.nfs/dfstab.nfs1-res
```

2. From one node in the Sun Cluster system, create the NFS resource group.

```
#scrgadm -a -g nfs-rg -y PathPrefix=/global/nfs
```

3. Add a logical host to the NFS resource group.

```
# scrgadm -a -L -g nfs-rg -l lh-nfs1
```

4. Configure the `HAStoragePlus` resource type.

Figure 188. Configuring the `HAStoragePlus` Resource Type

```
# scrgadm -c -g nfs-rg -h scnode-A,scnode-B
# scrgadm -a -g nfs-rg -j qfsnfs1-res -t SUNW.HAStoragePlus \
        -x FilesystemMountPoints=/global/qfsnfs1 \
        -x FilesystemCheckCommand=/bin/true
```

5. Bring the resource group online.

```
# scswitch -Z -g nfs-rg
```

6. Configure the NFS resource type and set a dependency on the `HAStoragePlus` resource.

```
# scrgadm -a -g nfs-rg -j nfs1-res -t SUNW.nfs -y \
Resource_dependencies=qfsnfs1-res
```

7. Bring the NFS resource online.

```
# scswitch -e -j nfs1-res
```

   The NFS resources `/net/lh-nfs1/global/qfsnfs1` is fully configured and highly available.

8. Before you announce the availability of the highly available NFS file system on the StorageTek QFS file system, validate that the resource group can be switched between all configured nodes without errors and taken online and offline.

Figure 189. Testing the Resource Group

```
# scswitch -z -g nfs-rg -h scnode-A
# scswitch -z -g nfs-rg -h scnode-B
# scswitch -F -g nfs-rg
# scswitch -Z -g nfs-rg
```

Changing the StorageTek QFS Configuration

This section demonstrates how to make changes to, disable, or remove the StorageTek QFS shared or unshared file system configuration. It contains the following sections:

- “To Change the Shared File System Configuration” on page 275
- “To Disable HA-NFS on a File System That Uses Raw Global Devices” on page 276
- “To Disable HA-NFS on a File System That Uses Solaris Volume Manager-Controlled Volumes” on page 277
- “To Disable HA-NFS on a StorageTek QFS File System That Uses VxVM-Controlled Volumes” on page 279

To Change the Shared File System Configuration

This procedure is based on the example in “Example Configuration” on page 230.

1. Log into each node as the `oracle` user and shut down the database instance and stop the listener.

Figure 190. Shutting Down the Database Instance and Listener

```
$ sqlplus “/as sysdba”
SQL > shutdown immediate
SQL > exit
$ lsnrctl stop listener
```

2. Log into the metadata server as `superuser` and bring the metadata server resource group into the unmanaged state.

Figure 191. Bringing the Resource Group Into an Unmanaged State

```
# scswitch -F -g qfs-rg
# scswitch -u -g qfs-rg
```

At this point, the shared file systems are unmounted on all nodes. You can now apply any changes to the file systems’ configuration, mount options, and so on. You can also re-create the file systems, if necessary. To use the file systems again after recreating them, follow the steps in “Example Configuration” on page 230.

If you want to make changes to the metadata server resource group configuration or to the StorageTek QFS software (For example, you might need to upgrade to new packages.), continue to Step 3.

3. As superuser, remove the resource, the resource group, and the resource type, and verify that everything is removed.

Figure 192. Disabling Resource Groups

```
# scswitch -n -j qfs-res
# scswitch -r -j qfs-res
# scrgadm -r -g qfs-rg
# scrgadm -r -t SUNW.qfs
# scstat
```

At this point, you can re-create the resource group to define different names, node lists, and so on. You can also remove or upgrade the StorageTek QFS shared software, if necessary. After the new software is installed, the metadata resource group and the resource can be recreated and can be brought online.

To Disable HA-NFS on a File System That Uses Raw Global Devices

Use this procedure to disable HA-NFS on an unshared StorageTek QFS file system that is using raw global devices. This example procedure is based on "Example 1" on page 241.

1. Use the `scswitch`(1M) `-F -g` command to take the resource group offline.

```
#scswitch -F -g nfs-rg
```

2. Disable the NFS, StorageTek QFS, and `LogicalHost` resource types.

Figure 193. Disabling the Resource Types

```
# scswitch -n -j nfs1-res
# scswitch -n -j qfsnfs1-res
# scswitch -n -j lh-nfs1
```

3. Remove the previously configured resources.

Figure 194. Removing the Resources

```
# scrgadm -r -j nfs1-res
# scrgadm -r -j qfsnfs1-res
# scrgadm -r -j lh-nfs1
```

4. Remove the previously configured resource group.

```
#scrgadm -r -g nfs-rg
```

5. Clean up the NFS configuration directories.

```
#rm -fr /global/nfs
```

6. Disable the resource types used, if they were previously added and are no longer needed.

Figure 195. Disabling the Resource Types That are no Longer Needed

```
# scrgadm -r -t SUNW.HAStoragePlus
# scrgadm -r -t SUNW.LogicalHostname
# scrgadm -r -t SUNW.nfs
```

To Disable HA-NFS on a File System That Uses Solaris Volume Manager-Controlled Volumes

Use this procedure to disable HA-NFS on an unshared StorageTek QFS file system that is using Solstice DiskSuite/Solaris Volume Manager-controlled volumes. This example procedure is based on "Example 2" on page 251.

1. Take the resource group offline.

```
#scswitch -F -g nfs-rg
```

2. Disable the NFS, StorageTek QFS, and `LogicalHost` resources types

Figure 196. Disabling the Resource Types

```
# scswitch -n -j nfs1-res
# scswitch -n -j qfsnfs1-res
# scswitch -n -j lh-nfs1
```

3. Remove the previously configured resources.

Figure 197. Removing the Previously Configured Resources

```
# scrgadm -r -j nfs1-res
# scrgadm -r -j qfsnfs1-res
# scrgadm -r -j lh-nfs1
```

4. Remove the previously configured resource group.

```
# scrgadm -r -g nfs-rg
```

5. Clean up the NFS configuration directories.

```
# rm -fr /global/nfs
```

6. Disable the resource types used, if they were previously added and are no longer needed.

Figure 198. Disabling the Resource Types

```
# scrgadm -r -t SUNW.HAStoragePlus
# scrgadm -r -t SUNW.LogicalHostname
# scrgadm -r -t SUNW.nfs
```

7. Delete RAID-5 and RAID-1 sets.

Figure 199. Deleting the RAID-5 and RAID-1 Sets

```
# metaclear -s nfsdg -f d30 d20 d21 d22 d23 d11 d1 d2 d3 d4
```

8. Remove mediation detection of drive errors.

Figure 200. Removing the Mediation Detection of Drive Errors

```
# metaset -s nfsdg -d  -m scnode-A
# metaset -s nfsdg -d  -m scnode-B
```

9. Remove the shared DID devices from the `nfsdg` disk group.

```
# metaset -s nfsdg -d -f /dev/did/dsk/d4 /dev/did/dsk/d5 \
        /dev/did/dsk/d6 /dev/did/dsk/d7
```

10. Remove the configuration of disk group `nfsdg` across nodes in the Sun Cluster system.

```
# metaset -s  nfsdg -d -f -h scnode-A scnode-B
```

11. Delete the metadatabase, if it is no longer needed.

Figure 201. Deleting the Metadatabase

```
# metadb -d -f /dev/dsk/c0t0d0s7
# metadb -d -f /dev/dsk/c1t0d0s7
# metadb -d -f /dev/dsk/c2t0d0s7
```

To Disable HA-NFS on a StorageTek QFS File System That Uses VxVM-Controlled Volumes

Use this procedure to disable HA-NFS on an unshared StorageTek QFS file system that is using VxVM-controlled volumes. This example procedure is based on "Example 3" on page 263.

1. Take the resource group offline.

```
# scswitch -F -g nfs-rg
```

2. Disable the NFS, StorageTek QFS, and `LogicalHost` resources types.

Figure 202. Disabling the Resource Types

```
# scswitch -n -j nfs1-res
# scswitch -n -j qfsnfs1-res
# scswitch -n -j lh-nfs1
```

3. Remove the previously configured resources.

Figure 203. Removing the Resources

```
# scrgadm -r -j nfs1-res
# scrgadm -r -j qfsnfs1-res
# scrgadm -r -j lh-nfs1
```

4. Remove the previously configured resource group.

```
# scrgadm -r -g nfs-rg
```

5. Clean up the NFS configuration directories.

```
# rm -fr /global/nfs
```

6. Disable the resource types used, if they were previously added and are no longer needed.

Figure 204. Disabling the Resource Types That are no Longer Needed

```
# scrgadm -r -t SUNW.HAStoragePlus
# scrgadm -r -t SUNW.LogicalHostname
# scrgadm -r -t SUNW.nfs
```

7. Delete the subdisk.

```
# vxdg destroy nfsdg
```

8. Remove the VxVM devices.

```
# vxdisk rm fabric_0 fabric_1 fabric_2 fabric_3 fabric_4
```

Advanced Topics

9

This chapter discusses advanced topics that are beyond the scope of basic system administration and usage. This chapter contains the following sections:

Daemons, Processes, and Tracing

It is useful to have an understanding of system daemons and processes when you are debugging. This section describes the StorageTek QFS and StorageTek ASM daemons and processes. It also provides information about daemon tracing.

Daemons and Processes

All StorageTek QFS and StorageTek ASM daemons are named in the form `sam-`*daemon_name*`d`, which is `sam-`, followed by the daemon name, and followed by the lowercase letter `d`. This convention allows the daemons to be identified easily. Processes are named in a similar manner; the difference is that they do not end in the lowercase letter `d`. Table 71. shows some of the daemons and processes that can be running on your system (others, such as `sam-genericd` and `sam-catserverd`, might also be running depending on system activities).

Table 71. Daemons and Processes

Process	Description
`sam-archiverd`	Automatically archives StorageTek ASM files. This process runs as long as the StorageTek ASM file system is mounted.
`sam-fsd`	Master daemon.
`sam-rftd`	Transfers data between multiple StorageTek ASM host systems.
`sam-robotsd`	Starts and monitors automated library media changer control daemons.
`sam-scannerd`	Monitors all manually mounted removable media devices. The scanner periodically checks each device for inserted archive media cartridges.
sam-sharefsd	Invokes the StorageTek QFS shared file system daemon.
`sam-releaser`	Attempts to release disk space occupied by previously archived files on StorageTek ASM file systems until a low water mark is reached. The releaser is started automatically when a high water mark is reached on disk cache and stops when it has finished releasing files. This is a process, not a daemon.
`sam-stagealld`	Controls the associative staging of StorageTek ASM files.
`sam-stagerd`	Controls the staging of StorageTek ASM files.
`sam-rpcd`	Controls the remote procedure call (RPC) application programming interface (API) server process.

When running StorageTek QFS or StorageTek ASM software, `init` starts the `sam-fsd` daemon as part of `/etc/inittab` processing. It is started at `init` levels 0, 2, 3, 4, 5, and 6. It should restart automatically in case of kill or failure.

When running StorageTek ASM software, the `sam-fsd` daemon creates the following processes:

- `sam-archiverd`. The `sam-archiverd` daemon starts the `sam-arcopy` and the `sam-arfind` processes.
- `sam-catserverd`. Issuing a `samd`(1M) `stop` command stops this daemon.
- `sam-rftd`.
- `sam-initd`.

- `sam-robotsd`. Issuing a `samd`(1M) `stop` command stops this daemon.
- `sam-scannerd`. Issuing a `samd`(1M) `stop` command stops this daemon.
- `sam-sharefsd`. One process is created for each StorageTek QFS shared file system.
- `sam-stagealld`.
- `sam-stagerd`.

Trace Files

Several StorageTek QFS and StorageTek ASM processes can write messages to trace files. These messages contain information about the state and progress of the work performed by the daemons. The messages are primarily used by StorageTek staff members to to improve performance and diagnose problems. The message content and format are subject to change from release to release.

Trace files can be used in debugging. Typically, trace files are not written. You can enable trace files for StorageTek ASM software by editing the `defaults.conf` file. You can enable tracing for all processes, or you can enable tracing for individual processes. For information about the processes that you can trace, see the `defaults.conf`(4) man page.

By default, the trace files are written to the `/var/opt/SUNWsamfs/trace` directory. In that directory, the trace files are named for the processes (`archiver`, `catserver`, `fsd`, `ftpd`, `recycler`, `sharefsd`, and `stager`). You can change the names of the trace files by specifying directives in the `defaults.conf` configuration file. You can also set a limit on the size of a trace file and rotate your tracing logs. For information about controlling tracing, see the `defaults.conf`(4) man page.

Trace File Content

Trace file messages contain the time and source of the message. The messages are produced by events in the processes. You can select the events by using directives in the `defaults.conf` file.

The default events are as follows:

- Customer notification `syslog` or `notify` file messages
- Nonfatal program errors
- Fatal `syslog` messages
- Process initiation and completion
- Other miscellaneous events

You can also trace the following events:

- Memory allocations
- Interprocess communication
- File actions
- Operator messages
- Queue contents when changed
- Other miscellaneous events

The default message elements (program name, process id (PID), and time) are always included and cannot be excluded. Optionally, the messages can also contain the following elements:

- The date. (The time is always included.)
- The source file name and line number.
- The event type.

Trace File Rotation

To prevent the trace files from growing indefinitely, the `sam-fsd` daemon monitors the size of the trace files and periodically executes the following command:

```
/opt/SUNWsamfs/sbin/trace_rotate
```

This script moves the trace files to sequentially numbered copies. You can modify this script to suit your operation. Alternatively, you can provide this function using `cron`(1) or some other facility.

Determining Which Processes Are Being Traced

To determine which processes are being traced currently, enter the `sam-fsd`(1M) command at the command line. Figure 205. shows the output from this command.

Figure 205. `sam-fsd`**(1M) Command Output**

```
# sam-fsd
Trace file controls:
sam-amld      /var/opt/SUNWsamfs/trace/sam-amld
              cust err fatal misc proc date
              size    0    age 0
sam-archiverd /var/opt/SUNWsamfs/trace/sam-archiverd
              cust err fatal misc proc date
              size    0    age 0
sam-catserverd /var/opt/SUNWsamfs/trace/sam-catserverd
              cust err fatal misc proc date
              size    0    age 0
```

Figure 205. `sam-fsd`**(1M) Command Output (Continued)**

```
sam-fsd       /var/opt/SUNWsamfs/trace/sam-fsd
              cust err fatal misc proc date
              size    0    age 0
sam-rftd      /var/opt/SUNWsamfs/trace/sam-rftd
              cust err fatal misc proc date
              size    0    age 0
sam-recycler  /var/opt/SUNWsamfs/trace/sam-recycler
              cust err fatal misc proc date
              size    0    age 0
sam-sharefsd  /var/opt/SUNWsamfs/trace/sam-sharefsd
              cust err fatal misc proc date
              size    0    age 0
sam-stagerd   /var/opt/SUNWsamfs/trace/sam-stagerd
              cust err fatal misc proc date
              size    0    age 0
sam-serverd   /var/opt/SUNWsamfs/trace/sam-serverd
              cust err fatal misc proc date
              size    0    age 0
sam-clientd   /var/opt/SUNWsamfs/trace/sam-clientd
              cust err fatal misc proc date
              size    0    age 0
sam-mgmt      /var/opt/SUNWsamfs/trace/sam-mgmt
              cust err fatal misc proc date
              size    0    age 0
License: License never expires.
```

For more information about enabling trace files, see the `defaults.conf`(4) man page and the `sam-fsd`(1M) man page.

■ Using the setfa(1) Command to Set File Attributes

The StorageTek QFS and StorageTek ASM file systems allow end users to set performance attributes for files and directories. Applications can enable these performance features on a per-file or per-directory basis. The following sections describe how the application programmer can use these features to select file attributes for files and directories, to preallocate file space, to specify the allocation method for the file, and to specify the disk stripe width.

For more information about implementing the features described in the following subsections, see the `setfa`(1) man page.

Selecting File Attributes for Files and Directories

File attributes are set using the `setfa`(1) command. The `setfa`(1) command sets attributes on a new or existing file. The file is created if it does not already exist.

You can set attributes on a directory as well as a file. When using `setfa`(1) with a directory, files and directories created within that directory inherit the attributes set in the original directory. To reset attributes on a file or directory to the default, use the `-d` (default) option. When the `-d` option is used, attributes are first reset to the default and then other attributes are processed.

Preallocating File Space

An end user can preallocate space for a file. This space is associated with a file so that no other files in the file system can use the disk addresses allocated to this file. Preallocation ensures that space is available for a given file, which avoids a file system full condition. Preallocation is assigned at the time of the request rather than when the data is actually written to disk.

Note that space can be wasted when preallocating files. If the file size is less than the allocation amount, the kernel allocates space to the file from the current file size up to the allocation amount. When the file is closed, space below the allocation amount is not freed.

You can preallocate space for a file by using the `setfa`(1) command with either the `-L` or the `-l` (lowercase letter L) options. Both options accept a file length as their argument. You can use the `-L` option for an existing file, and that file either can be empty or it can contain data. Use the `-l` option for a file that has no data yet. If you use the `-l` option, the file cannot grow beyond its preallocated limit.

For example, to preallocate a 1-gigabyte file named `/qfs/file_alloc`, type the following:

```
# setfa -l 1g /qfs/file_alloc
```

After space for a file has been preallocated, truncating a file to 0 length or removing the file returns all space allocated for a file. There is no way to return only part of a file’s preallocated space to the file system. In addition, if a file is preallocated in this manner, there is no way to extend the file beyond its preallocated size in future operations.

Selecting a File Allocation Method and Stripe Width

By default, a file created uses the allocation method and stripe width specified at mount time (see the `mount_samfs`(1M) man page). However, an end user might want to use a different allocation scheme for a file or directory of files, and this can be accomplished by using the `setfa`(1) command with the `-s` (stripe) option.

The allocation method can be either round-robin or striped. The `-s` option determines the allocation method and the stripe width, and Table 72. shows the effect of this option.

Table 72. File Allocations and Stripe Widths

`-s` *stripe*	**Allocation Method**	**Stripe Width**	**Explanation**
0	Round-robin	n/a	The file is allocated on one device until that device has no space.
1-255	Striped	1-255 DAUs	The file stripes across all disk devices with this number of DAUs per disk.

The following example shows how to create a file explicitly by specifying a round-robin allocation method:

```
# setfa -s 0 /qfs/100MB.rrobin
```

The following example shows how to create a file explicitly by specifying a striped allocation method with a stripe width of 64 DAUs (preallocation is not used):

```
# setfa -s 64 /qfs/file.stripe
```

Selecting a Striped Group Device

Striped group devices are supported for StorageTek QFS file systems only.

A user can specify that a file begin allocation on a particular striped group. If the file allocation method is round-robin, the file is allocated on the designated stripe group.

Figure 206. shows `setfa`(1) commands that specify that `file1` and `file2` be independently spread across two different striped groups.

Figure 206. `setfa`**(1) Commands to Spread Files Across Striped Groups**

```
# setfa -g0 -s0 file1
# setfa -g1 -s0 file2
```

This capability is particularly important for applications that must achieve levels of performance that approach raw device speeds. For more information, see the `setfa`(1) man page.

Accommodating Large Files

When manipulating very large files, pay careful attention to the size of disk cache available on the system. If you try to write a file that is larger than your disk cache, behavior differs depending on the type of file system you are using, as follows:

- If you are using the StorageTek QFS file system, the system returns an `ENOSPC` error.
- If you are using the StorageTek ASM file system, the program blocks, waiting for space that might never exist, because there is not enough disk space available to handle such requests.

  If you are operating within a StorageTek ASM environment and if your application requires writing a file that is larger than the disk cache, you can segment the file using the `segment`(1) command. For more information about the `segment`(1) command, see the `segment`(1) man page or see the *StorageTek ASM Storage and Archive Manager Guide*.

Multireader File System

The multireader file system consists of a single writer host and multiple reader hosts. The `writer` and `reader` mount options that enable the multireader file system are compatible with StorageTek QFS file systems only. The mount options are described in this section and on the `mount_samfs`(1M) man page.

You can mount the multireader file system on the single writer host by specifying the `-o writer` option on the `mount`(1M) command. The host system with the `writer` mount option is the only host system that is allowed to write to the file system. The `writer` host system updates the file system. You must ensure that only one host in a multireader file system has the file system mounted with the `writer` mount option enabled. If `-o writer` is specified, directories are written through to disk at each change and files are written through to disk at close.

CAUTION: The multireader file system can become corrupted if more than one writer host has the file system mounted at one time. It is the site’s responsibility to insure that this situation does not occur.

You can mount a multireader file system on one or more reader hosts by specifying the `-o reader` option on the `mount`(1M) command. There is no limit to the number of host systems that can have the multireader file system mounted as a reader.

A major difference between the multireader file system and StorageTek QFS shared file system is that the multireader host read metadata from the disk, and the client hosts of a StorageTek QFS shared file system read metadata over the network. The StorageTek QFS shared file system supports multireader hosts. In this configuration, multiple shared hosts can be adding content while multiple reader hosts can be distributing content.

Note: You cannot specify the `writer` option on any host if you are mounting the file system as a StorageTek QFS shared file system. You can, however, specify the `reader` option.

If you want a StorageTek QFS shared file system client host to be a read-only host, mount the StorageTek QFS shared file system on that host with both the `shared` and `reader` mount options. In addition, set the `sync_meta` mount option to 1 if you use the `reader` option in a StorageTek QFS shared file system. For more information about the StorageTek QFS shared file system, see "StorageTek QFS Shared File System" on page 85. For more information about mount options, see the `mount_samfs`(1M) man page.

You must ensure that all readers in a multireader file system have access to the device definitions that describe the `ma` device. Copy the lines from the `mcf` file that resides on the primary metadata server host to the `mcf` files on the alternate metadata servers. After copying the lines, you might need to update the information about the disk controllers because depending on your configuration, disk partitions might not show up the same way across all hosts.

In a multireader file system environment, the StorageTek QFS software ensures that all servers that access the same file system can always access the current environment. When the writer closes a file, the StorageTek QFS file system writes all information for that file to disk immediately. A `reader` host can access a file after the file is closed by the writer. You can specify the `refresh_at_eof` mount option to help ensure that no host system in a multireader file system risks getting out of sync with the file system.

By default, the metadata information for a file on a `reader` host is invalidated and refreshed every time a file is accessed. If the data changed, it is invalidated. This includes any type of access, whether through `cat`(1), `ls`(1), `touch`(1), `open`(2), or other methods. This immediate refresh rate ensures that the data is correct at the time the refresh is done, but it can affect performance. Depending on your site preferences, you can use the `mount`(1M) command's `-o invalid=`*n* option to specify a refresh rate between 0 seconds and 60 seconds. If the refresh rate is set to a small value, the StorageTek QFS file system reads the directory and other metadata information *n* seconds after the last refresh. More frequent refreshes result in more overhead for the system, but stale information can exist if *n* is nonzero.

CAUTION: If a file is open for a read on a `reader` host, there is no protection for that file being removed or truncated by the writer. You must use another mechanism, such as application locking, to protect the reader from inadvertent writer actions.

Using the SAN-QFS File System in a Heterogeneous Computing Environment

The SAN-QFS file system enables multiple hosts to access the data stored in a StorageTek QFS system at full disk speeds. This capability can be especially useful for database, data streaming, web page services, or any application that demands high-performance, shared-disk access in a heterogeneous environment.

You can use the SAN-QFS file system in conjunction with fiber-attached devices in a storage area network (SAN). The SAN-QFS file system enables high-speed access to data through StorageTek QFS software and software such as Tivoli SANergy file sharing software. To use the SAN-QFS file system, you must have both the SANergy (2.2.4 or later) and the StorageTek QFS software. For information about the levels of StorageTek QFS and SANergy software that are supported, contact your StorageTek sales representative.

Note: In environments that include Solaris Operating Systems (OS), use the StorageTek QFS shared file system, not the SAN-QFS file system, on the Solaris hosts.

For information about the StorageTek QFS shared file system, see the "StorageTek QFS Shared File System" on page 85.

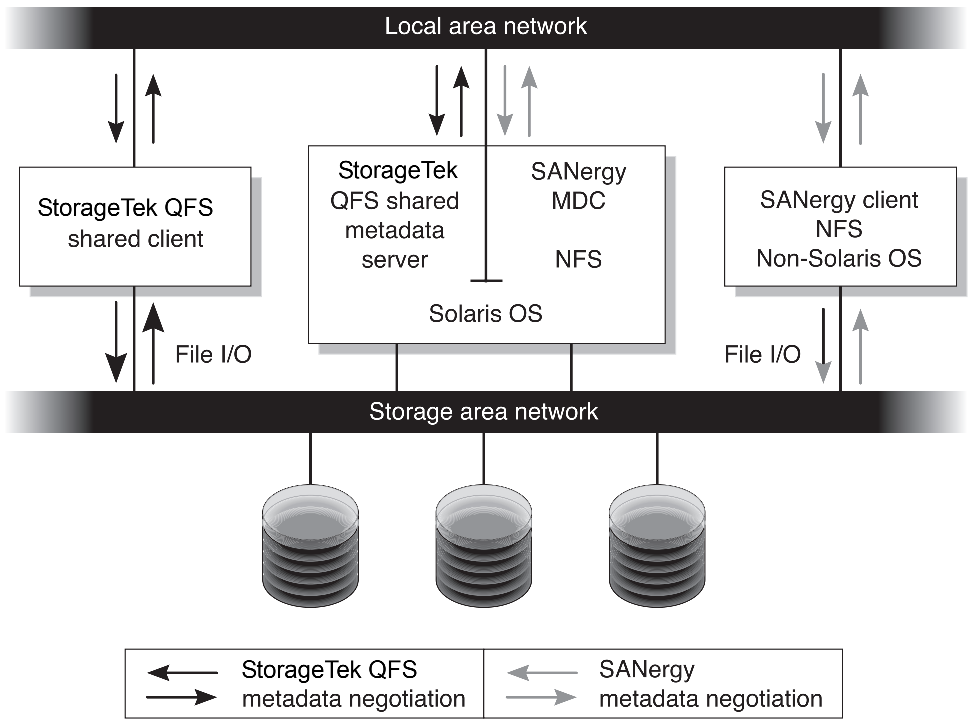
For a comparison of the StorageTek QFS shared file system to the SAN-QFS file system, see "SAN-QFS Shared File System and StorageTek QFS Shared File System Comparison" on page 297.

The following sections describe other aspects of the SAN-QFS file system:

Figure 207. depicts a SAN-QFS file system that uses both the StorageTek QFS software and the SANergy software and shows that the clients and the MDC system manage metadata across the LAN. The clients perform I/O directly to and from the storage devices.

Note that all clients running only the Solaris OS are hosting the StorageTek QFS software, and that all heterogeneous clients running an OS other than Solaris are hosting the SANergy software and the NFS software. The SAN-QFS file system's metadata server hosts both the StorageTek QFS and the SANergy software. This server acts not only as the metadata server for the file system but also as the SANergy metadata controller (MDC).

Figure 207. SAN-QFS File System Using StorageTek QFS Software and SANergy Software.



Note: This documentation assumes that your non-Solaris clients are hosting SANergy software and NFS software for file system sharing. The text and examples in this document reflect this configuration. If your non-Solaris clients host the Samba software instead of the NFS software, see your Samba documentation.

Before You Begin

Before you enable the SAN-QFS file system, keep the following configuration considerations in mind and plan accordingly:

- Disks configured for use in a SAN-QFS file system cannot be under the control of a volume manager.
- To enable or to relocate the StorageTek QFS metadata server in a SAN-QFS environment, the new metadata server system must be configured as a SANergy meta data controller.

Enabling the SAN-QFS File System

The following procedures describe how to enable the SAN-QFS file system. Perform these procedures in the order in which they are presented:

- "To Enable the SAN-QFS File System on the Metadata Controller" on page 292.
- "To Enable the SAN-QFS File System on the Clients" on page 293..
- "To Install the SANergy Software on the Clients" on page 293.

To Enable the SAN-QFS File System on the Metadata Controller

When you use the SAN-QFS file system, one host system in your environment acts as the SANergy metadata controller (MDC). This is the host system upon which the StorageTek QFS file system resides.

1. Log in to the host upon which the StorageTek QFS file system resides and become `superuser`.
2. Verify that the StorageTek QFS file system is tested and fully operational.
3. Install and configure the SANergy software.

   For instructions, see your SANergy documentation.
4. Use the `pkginfo`(1) command to verify the SANergy software release level.

```
# pkginfo -l SANergy
```

5. Ensure that the file system is mounted.

   Use the `mount`(1M) command either to verify the mount or to mount the file system.
6. Use the `share`(1M) command in the following format to enable NFS access to client hosts.

```
MDC# share -F nfs -d qfs_file_system_name /mount_point
```

   For *qfs_file_system_name*, specify the name of your StorageTek QFS file system, such as, `qfs1`. For more information about the `share`(1M) command, see the `share`(1M) or `share_nfs`(1M) man page.

   For *mount_point*, specify the mount point of *qfs_file_system_name*.
7. If you're connecting to Microsoft Windows clients, configure Samba, rather than NFS, to provide security and namespace features.

To do this, add the `SANERGY_SMBPATH` environment variable in the `/etc/init.d/sanergy` file.

8. Edit the file system table (`/etc/dfs/dfstab`) on the MDC to enable access at boot time. (Optional)

   Perform this step if you want to automatically enable this access at boot time.

9. Proceed to "To Enable the SAN-QFS File System on the Clients" on page 293.

To Enable the SAN-QFS File System on the Clients

The SAN-QFS file system supports several client hosts. For example, this file system supports IRIX, Windows, AIX, and LINUX hosts. For information about the specific clients supported, see your StorageTek sales representative.

Every client has different operational characteristics. This procedure uses general terms to describe the actions you must take to enable the SAN-QFS file system on the clients. For information specific to your clients, see the documentation provided with your client hosts.

1. Log in to each of the client hosts.

2. Edit the file system defaults table on each client and add the file system.

   For example, on a Solaris OS, edit the `/etc/vfstab` file on each client and add *qfs_file_system_name* from Step 6 to the table, as follows:

```
server:/qfs1  -  /qfs1  nfs  -  yes
noac,hard,intr,timeo=1000
```

   On other operating system platforms, the file system defaults table might reside in a file other than `/etc/vfstab`. For example, on LINUX systems, this file is `/etc/fstab`.

   For more information about editing the `/etc/vfstab` file, see *StorageTek ASM Installation and Configuration Guide*. For information about required or suggested NFS client mount options, see your SANergy documentation.

3. Proceed to "To Install the SANergy Software on the Clients" on page 293.

To Install the SANergy Software on the Clients

The following procedure describes the SANergy installation process in general terms. For platform-specific information, consult your SANergy documentation.

1. Install and configure the SANergy software.

   For instructions, see your SANergy documentation.

2. Use the `mount` command to NFS mount the file system.

   Use this command with one of the following lists of mount options:

   Option list 1:

```
# mount -o acregmin=0, acregmax=0, actimeo=0, noac \
host:/mount_point/ local_mount_point
```

Option list 2:

```
# mount -noac host:/mount_point/ local_mount_point
```

For *host*, specify the MDC.

For *mount_point*, specify the mount point of the StorageTek QFS file system on the MDC.

For *local_mount_point*, specify the mount point on the SANergy client.

3. Use the SANergy `fuse` command to fuse the software.

   For example:

```
# fuse|mount_point
```

For *mount_point*, specify the mount point on the SANergy client.

Unmounting the SAN-QFS File System

The following procedures describe how to unmount a SAN-QFS file system that is using the SANergy software. Perform these procedures in the order in which they are presented:

- "To Unmount the SAN-QFS File System on the SANergy Clients" on page 294
- "To Unmount the SAN-QFS File System on the Metadata Controller" on page 295
- "To Unmount the SAN-QFS File System on the StorageTek QFS Clients" on page 295
- "To Unmount the SAN-QFS File System on the StorageTek QFS Server" on page 296

To Unmount the SAN-QFS File System on the SANergy Clients

Follow these steps for each client host on which you want to unmount the SAN-QFS file system.

1. Log in to the client host and become `superuser`.
2. Use the SANergy `unfuse` command to unfuse the file system from the software.

```
# unfuse|mount_point
```

   For *mount_point*, specify the mount point on the SANergy client.

3. Use the `umount`(1M) command to unmount the file system from NFS.

```
# umount host:/mount_point/ local_mount_point
```

For *host*, specify the MDC.

For *mount_point*, specify the mount point of the StorageTek QFS file system on the MDC.

For *local_mount_point*, specify the mount point on the SANergy client.

To Unmount the SAN-QFS File System on the Metadata Controller

1. Log in to the MDC system and become `superuser`.
2. Use the `unshare`(1M) command to disable NFS access to client hosts.

```
MDC# unshare qfs_file_system_name /mount_point
```

   For *qfs_file_system_name*, specify the name of your StorageTek QFS file system, such as, `qfs1`. For more information about the un`share`(1M) command, see the un`share`(1M) man page.

   For *mount_point*, specify the mount point of *qfs_file_system_name*.

To Unmount the SAN-QFS File System on the StorageTek QFS Clients

Follow these steps on each participating client host.

1. Log in to a StorageTek QFS client host and become `superuser`.
2. Use the `umount`(1M) command to unmount the file system.

```
# umount /qfs1
```

To Unmount the SAN-QFS File System on the StorageTek QFS Server

1. Log in to the host system upon which the StorageTek QFS file system resides and become `superuser`.
2. Use the `umount`(1M) command to unmount the file system.

Troubleshooting: Unmounting a SAN-QFS File System with SANergy File Holds

SANergy software issues holds on StorageTek QFS files to reserve them temporarily for accelerated access. If SANergy crashes when holds are in effect, you will not be able to unmount the file system. If you are unable to unmount a SAN-QFS file system, examine the `/var/adm/messages` file and look for console messages that describe outstanding SANergy holds.

Whenever possible, allow the SANergy file-sharing function to clean up its holds, but in an emergency, or in case of a SANergy File Sharing system failure, use the `samunhold`(1M) command and the following procedure to avoid a reboot.

To Unmount a File System in the Presence of SANergy File Holds

1. Use the `unshare`(1M) command to disable NFS access.
2. Use the `samunhold`(1M) command to release the SANergy file system holds.

   For more information about this command, see the `samunhold`(1M) man page.
3. Use the `umount`(1M) command to unmount the file system.

Block Quotas in a SAN-QFS File System

The SANergy software does not enforce block quotas. Therefore, it is possible for you to exceed a block quota when writing a file with the SANergy software. For more information on quotas, see "Enabling Quotas" on page 202.

File Data and File Attributes in a SAN-QFS File System

The SANergy software uses the NFS software for metadata operations, which means that the NFS close-to-open consistency model is used for file data and attributes. File data and attributes are not coherent among SANergy clients for open files.

Using samgrowfs(1M) to Expand SAN-QFS File Systems

You can use the `samgrowfs`(1M) command to increase the size of a SAN-QFS file system. To perform this task, follow the procedures described in "Adding Disk Cache to a File System" on page 73. When using this procedure, be aware that the line-by-line device order in the `mcf` file must match the order of the devices listed in the file system's superblock. The devices listed in the file system's superblock are numbered in the order encountered in the `mcf` file (when created).

When the `samgrowfs`(1M) command is issued, the devices that had been in the `mcf` file prior to issuing the `samgrowfs`(1M) command keep their position in the superblock. New devices are written to subsequent entries in the order encountered.

If this new order does not match the order in the superblock, the SAN-QFS file system cannot be fused.

SAN-QFS Shared File System and StorageTek QFS Shared File System Comparison

The SAN-QFS file system and the StorageTek QFS shared file system are both shared file systems with the following similarities:

- Both can stage files.
- Both are useful in data capture environments in which it is desirable that the primary file system host not be responsible for writing the data.
- Both are advantageous in environments where there is contention for writing files.

Table 73. shows the file systems differences.

Table 73. SAN-QFS Shared File System Versus StorageTek QFS Shared File System

SAN-QFS File System	StorageTek QFS Shared File System
Uses NFS protocol for metadata.	Uses natural metadata.
Preferred in heterogeneous computing environments (that is, when not all hosts are Sun systems).	Preferred in homogeneous Solaris OS environments.
Useful in environments where multiple, heterogeneous hosts must be able to write data.	Multiple hosts can write. Preferred when multiple hosts must write to the same file at the same time.

I/O Performance

The StorageTek QFS and StorageTek ASM file systems support paged I/O, direct I/O, and switching between the I/O types. The following sections describe these I/O types.

Paged I/O

Paged I/O (also called buffered or cached I/O) is selected by default.

Direct I/O

Direct I/O is a process by which data is transferred directly between the user's buffer and the disk. This means that much less time is spent in the system. For performance purposes, specify direct I/O only for large, block-aligned, sequential I/O.

The `setfa`(1) command and the `sam_setfa`(3) library routine both have a `-D` option that sets the direct I/O attribute for a file and/or directory. If applied to a directory, files and directories created in that directory inherit the direct I/O attribute. After the `-D` option is set, the file uses direct I/O.

You can also select direct I/O for a file by using the Solaris OS `directio`(3C) function call. If you use the function call to enable direct I/O, it is a temporary setting. The setting lasts only while the file is active.

To enable direct I/O on a file-system basis, do one of the following:

- Specify the `-o forcedirectio` option on the `mount`(1M) command.
- Put the `forcedirectio` keyword in the mount option column of the `/etc/vfstab` file; or use it as a directive in the `samfs.cmd` file.

For more information, see the `setfa`(1), `sam_setfa`(3), `directio`(3C), `samfs.cmd`(4) , and `mount_samfs`(1M) man pages.

I/O Switching

The StorageTek QFS and StorageTek ASM file systems support automatic I/O switching. I/O switching is a process by which you can specify that a certain amount of paged I/O should occur before the system switches to direct I/O. This automatic, direct I/O switching allows the system to perform a site-defined amount of consecutive I/O operations and then automatically switch from paged I/O to direct I/O. By default, paged I/O is performed, and I/O switching is disabled.

I/O switching should reduce page cache usage on large I/O operations. To enable this, use the `dio_wr_consec` and `dio_rd_consec` parameters as directives in the `samfs.cmd` file or as options to the `mount`(1M) command. You can also enable this by using `samu`(1M).

For more information about these options, see the `mount_samfs`(1M) or `samfs.cmd`(4) man pages.

Increasing Large File Transfer Performance

StorageTek QFS and StorageTek ASM file systems are tuned to work with a mix of file sizes. You can increase the performance of disk file transfers for large files by enabling file system settings.

Note: StorageTek recommends that you experiment with performance tuning outside of a production environment. Tuning these variables incorrectly can have unexpected effects on the overall system.

To Increase File Transfer Performance

1. Set the maximum device read/write directive.

   The `maxphys` parameter in the Solaris `/etc/system` file controls the maximum number of bytes that a device driver reads or writes at any one time. The default value for the `maxphys` parameter can differ depending on the level of your Sun Solaris OS, but it is typically around 128 kilobytes.

Add the following line to `/etc/system` to set `maxphys` to 8 megabytes:

```
set maxphys = 0x800000
```

2. Set the SCSI disk maximum transfer parameter.

   The `sd` driver enables large transfers for a specific file by looking for the `sd_max_xfer_size` definition in the `/kernel/drv/sd.conf` file. If it is not defined, it uses the value defined in the `sd` device driver definition, `sd_max_xfer_size`, which is 1024*1024 bytes.

   To enable and encourage large transfers, add the following line at the end of the `/kernel/drv/sd.conf` file:

```
sd_max_xfer_size=0x800000;
```

3. Set the fibre disk maximum transfer parameter.

   The `ssd` driver enables large transfers for a specific file by looking for the `ssd_max_xfer_size` definition in the `/kernel/drv/ssd.conf` file. If it is not defined, it uses the value defined in the `ssd` device driver definition, `ssd_max_xfer_size`, which is 1024*1024 bytes.

   Add the following line at the end of the `/kernel/drv/ssd.conf` file:

```
ssd_max_xfer_size=0x800000;
```

4. Reboot the system.
5. Set the `writebehind` parameter.

   This step affects paged I/O only.

   The `writebehind` parameter specifies the number of bytes that are written behind by the file system when performing paged I/O on a StorageTek QFS or StorageTek ASM file system. Matching the `writebehind` value to a multiple of the RAID's read-modify-write value can increase performance.

   This parameter is specified in units of kilobytes and is truncated to an 8-kilobyte multiple. If set, this parameter is ignored when direct I/O is performed. The default `writebehind` value is 512 kilobytes. This value favors large-block, sequential I/O.

   Set the `writebehind` size to a multiple of the RAID 5 stripe size for both hardware and software RAID 5. The RAID 5 stripe size is the number of data disks multiplied by the configured stripe width.

   For example, assume that you configure a RAID 5 device with three data disks plus one parity disk (3+1) with a stripe width of 16 kilobytes. The `writebehind` value should be 48 kilobytes, 96 kilobytes, or some other multiple, to avoid the overhead of the read-modify-write RAID 5 parity generation.

   For StorageTek QFS file systems, the DAU (`sammkfs`(1M) `-a` command) should also be a multiple of the RAID 5 stripe size. This allocation ensures that the blocks are contiguous.

   You should test the system performance after resetting the `writebehind` size. The following example shows testing timings of disk writes:

```
# timex dd if=/dev/zero of=/sam/myfile bs=256k count=2048
```

   You can set `writebehind` parameter from a mount option, from within the `samfs.cmd` file, from within the `/etc/vfstab` file, or from a command within the `samu`(1M) utility. For information about enabling this from a mount option, see the `-o writebehind=`*n* option on the `mount_samfs`(1M) man page. For information about enabling this from the `samfs.cmd` file, see the `samfs.cmd`(4) man page. For information about enabling this from within `samu`(1M), see the `samu`(1M) man page.

6. Set the `readahead` parameter.

   This step affects paged I/O only.

   The `readahead` parameter specifies the number of bytes that are read ahead by the file system when performing paged I/O on a StorageTek QFS or StorageTek ASM file system. This parameter is specified in units of kilobytes and is truncated to an 8-kilobyte multiple. If set, this parameter is ignored when direct I/O is performed.

Increasing the size of the `readahead` parameter increases the performance of large file transfers, but only to a point. You should test the performance of the system after resetting the `readahead` size until you see no more improvement in transfer rates. The following is an example method of testing timings on disk reads:

```
# timex dd if=/sam/myfile of=/dev/null bs=256k
```

The `readahead` parameter should be set to a size that increases the I/O performance for paged I/O. Also note that too large a `readahead` size can hurt performance. You should test various `readahead` sizes for your environment. It is important to consider the amount of memory and number of concurrent streams when you set the `readahead` value. Setting the `readahead` value multiplied by the number of streams to a value that is greater than memory can cause page thrashing.

The default `readahead` is 1024 kilobytes. This value favors large-block, sequential I/O. For short-block, random I/O applications, set `readahead` to the typical request size. Database applications do their own `readahead`, so for these applications, set `readahead` to 0.

The `readahead` setting can be enabled from a mount option, from within the `samfs.cmd` file, from within the `/etc/vfstab` file, or from a command within the `samu`(1M) utility. For information about enabling this from a mount option, see the `-o readahead=`*n* option on the `mount_samfs`(1M) man page. For information about enabling this from the `samfs.cmd` file, see the `samfs.cmd`(4) man page. For information about enabling this from within `samu`(1M), see the `samu`(1M) man page.

7. Set the stripe width.

   The `-o stripe=`*n* option on the `mount`(1M) command specifies the stripe width for the file system. The stripe width is based on the disk allocation unit (DAU) size. The *n* argument specifies that *n* * DAU bytes are written to one device before switching to the next device. The DAU size is set when the file system is initialized by the `sammkfs`(1M) `-a` command.

   If `-o stripe=0` is set, files are allocated to file system devices using the round-robin allocation method. Each file is created on the next device. Each file is completely allocated on this device until that device is full. Round robin is the preferred setting for a multistream environment. If `-o stripe=`*n* is set to an integer greater than 0, files are allocated to file system devices using the stripe method. To determine the appropriate `-o stripe=`*n* setting, try varying the setting and taking performance readings. Striping is the preferred setting for turnkey applications with a required bandwidth.

You can also set the stripe width from the /etc/vfstab file or from the samfs.cmd file.

For more information about the mount(1M) command, see the mount_samfs(1M) man page. For more information about the samfs.cmd file, see the samfs.cmd(4) man page.

Qwrite

The Qwrite capability can be enabled in StorageTek QFS environments.

By default, the StorageTek QFS file systems disable simultaneous reads and writes to the same file. This is the mode defined by the UNIX vnode interface standard, which gives exclusive access to only one write while other writers and readers must wait. Qwrite enables simultaneous reads and writes to the same file from different threads.

The Qwrite feature can be used in database applications to enable multiple simultaneous transactions to the same file. Database applications typically manage large files and issue simultaneous reads and writes to the same file. Unfortunately, each system call to a file acquires and releases a read/write lock inside the kernel. This lock prevents overlapped (or simultaneous) operations to the same file. If the application itself implements file locking mechanisms, the kernel locking mechanism impedes performance by unnecessarily serializing I/O.

Qwrite can be enabled in the /etc/vfstab file, in the samfs.cmd file, and as a mount option. The -o qwrite option on the mount(1M) command bypasses the file system locking mechanisms (except for applications accessing the file system through NFS) and lets the application control data access. If qwrite is specified, the file system enables simultaneous reads and writes to the same file from different threads. This option improves I/O performance by queuing multiple requests at the drive level.

The following example uses the mount(1M) command to enable Qwrite on a database file system:

```
# mount -F samfs -o qwrite /db
```

For more information about this feature, see the qwrite directive on the samfs.cmd(4) man page or the -o qwrite option on the mount_samfs(1M) man page.

Setting the Write Throttle

By default, the StorageTek QFS and StorageTek ASM file systems set the -o wr_throttle=*n* option to the mount(1M) command to 16 megabytes. The -o wr_throttle=*n* option limits the number of outstanding write kilobytes for one file to *n*.

If a file has *n* write kilobytes outstanding, the system suspends an application that attempts to write to that file until enough bytes have completed the I/O to allow the application to be resumed.

If your site has thousands of streams, such as thousands of NFS-shared workstations accessing the file system, you can tune the `-o wr_throttle=`*n* option in order to avoid memory stales. Generally, the number of streams multiplied by 1024 x the *n* argument to the `-o wr_throttle=`*n* option should be less than the total size of the host system's memory minus the memory needs of the Solaris OS. In other words:

```
number_of_streams * n * 1024 < total_memory - Solaris_OS_memory_needs
```

For turnkey applications, you might want to use a size larger than the default 16,384 kilobytes because this keeps more pages in memory.

Setting the Flush-Behind Rate

Two mount parameters control the flush-behind rate for pages written sequentially and stage pages. The `flush_behind` and `stage_flush_behind` mount parameters are read from the `samfs.cmd` file, the `/etc/vfstab` file, or from the `mount`(1M) command.

The `flush_behind=`*n* mount parameter sets the maximum flush-behind value. Modified pages that are being written sequentially are written to disk asynchronously to help the Sun Solaris VM layer keep pages clean. To enable this feature, set *n* to be an integer, $16 \le n \le 8192$. By default, *n* is set to 0, which disables this feature. The *n* argument is specified in kilobyte units.

The `stage_flush_behind=`*n* mount parameter sets the maximum stage flush-behind value. Stage pages that are being staged are written to disk asynchronously to help the Sun Solaris VM layer keep pages clean. To enable this feature, set *n* to be an integer such that, $16 \le n \le 8192$. By default, *n* is set to 0, which disables this feature. The *n* argument is specified in kilobyte units.

For more information about these mount parameters, see the `mount_samfs`(1M) man page or the `samfs.cmd`(4) man page.

Tuning the Number of Inodes and the Inode Hash Table

The StorageTek QFS and StorageTek ASM file systems allow you to set the following two tunable parameters in the `/etc/system` file:

- `ninodes`
- `nhino`

To enable nondefault settings for these parameters, edit the `/etc/system` file, and then reboot your system.

The following sections describe these parameters in more detail.

The ninodes Parameter

The `ninodes` parameter specifies the maximum number of default inodes. The value for `ninodes` determines the number of in-core inodes that StorageTek QFS and StorageTek ASM keep allocated to themselves, even when applications are not using many inodes.

The format for this parameter in the `/etc/system` file is as follows:

```
set samfs:ninodes = value
```

The range for *value* is `16` ≤ *value* ≤ `2000000`. The default *value* for `ninodes` is one of the following:

- A *value* that is equal to the `ncsize` setting. The `ncsize` parameter is a Solaris tuning parameter that specifies the number of entries in the directory name look-up cache (DNLC). For more information about `ncsize`, see the *Solaris Tunable Parameters Reference Manual*.
- `2000`. The file systems set `ninodes` to `2000` if the `ncsize` setting is zero or out of range.

  For example:

```
set samfs:ninodes = 4000
```

The nhino Parameter

The `nhino` parameter specifies the size of the in-core inode hash table.

The format for this parameter in the `/etc/system` file is as follows:

```
set samfs:nhino = value
```

The range for *value* is `1` ≤ *value* ≤ `1048756`. *value* must be a nonzero power of two. The default *value* for `nhino` is one of the following:

- A value that is equal to the ninodes value divided by eight and then, if necessary, rounded up to the nearest power of two. For example, assume that the following line exists in `/etc/system`:

```
set samfs:ninodes 8000
```

For this example, if `nhino` is not set, the system assumes `1024`, which is `8000` divided by 8 and then rounded up to the nearest power of two.

- 512. The file systems set `nhino` to `512` if the `ninodes` setting is out of range.

  For example:

```
set samfs:nhino = 1024
```

When to Set the ninodes and nhino Parameters

When searching for an inode by number (after obtaining an inode number from a directory or after extracting an inode number from an NFS file handle), the StorageTek QFS and StorageTek ASM file systems search their cache of in-core inodes. To speed this process, they maintain a hash table to decrease the number of inodes they must check.

A larger hash table reduces the number of comparisons and searches, at a modest cost in memory usage. If the `nhino` value is too large, the system is slower when undertaking operations that sweep through the entire inode list (inode syncs and unmounts). For sites that manipulate large numbers of files and sites that do extensive amounts of NFS I/O, it can be advantageous to set these parameter values to larger than the defaults.

If your site has file systems that contain only a small number of files, it might be advantageous to make these numbers smaller than the defaults. This could be the case, for example, if you have a file system into which you write large single-file `tar`(1) files to back up other file systems.

Glossary

A

addressable storage The storage space encompassing online, nearline, offsite, and offline storage that is user-referenced through a StorageTek QFS or StorageTek ASM file system.

archive media The media to which an archive file is written. Archive media can be removable tape or magneto-optical cartridges in a library. In addition, archive media can be a mount point on another system.

archive storage Copies of file data that have been created on archive media.

archiver The archive program that automatically controls the copying of files to removable cartridges.

audit (full) The process of loading cartridges to verify their VSNs. For magneto-optical cartridges, the capacity and space information is determined and entered into the automated library's catalog.

automated library A robotically controlled device designed to automatically load and unload removable media cartridges without operator intervention. An automated library contains one or more drives and a transport mechanism that moves cartridges to and from the storage slots and the drives.

B

backup storage A snapshot of a collection of files for the purpose of preventing inadvertent loss. A backup includes both the file's attributes and associated data.

block allocation map A bitmap representing each available block of storage on a disk and indicating whether the block is in use or free.

block size See DAU.

C

cartridge A physical entity that contains media for recording data. A tape or optical disk. Sometimes referred to as *a piece of media*, *a volume*, or *the medium*.

catalog A record of the VSNs in an automated library. There is one catalog for each automated library, and at a site, there is one historian for all automated libraries.

client-server The model of interaction in a distributed system in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called the client. The program satisfying the response is called the server.

connection The path between two protocol modules that provides reliable stream delivery service. A TCP connection extends from a TCP module on one machine to a TCP module on the other.

D

data device For a StorageTek QFS or StorageTek ASM file system, a device or group of devices upon which file data is stored.

DAU (Disk allocation unit) The basic unit of online storage. Also called block size.

In addition, the StorageTek QFS file systems support a fully adjustable DAU, sized from 16 kilobytes through 65,528 kilobytes. The

DAU you specify must be a multiple of 8 kilobytes.

The StorageTek ASM file systems support both a small and a large DAU. The small DAU is 4 kilobytes (2^{14} or 4096 bytes). The large DAU is 16, 32, or 64 kilobytes. The available DAU size pairs are 4 and 16; 4 and 32; and 4 and 64.

device logging A configurable feature that provides device-specific error information used to analyze device problems.

device scanner Software within the StorageTek ASM file system that periodically monitors the presence of all manually mounted removable devices and that detects the presence of mounted cartridges that can be requested by a user or other process.

direct access A file attribute (stage never) designating that a nearline file can be accessed directly from the archive media and need not be retrieved to disk cache.

direct-attached library An automated library connected directly to a server using a SCSI interface. A SCSI attached library is controlled directly by the StorageTek ASM software by using the SCSI standard for automated libraries.

direct I/O An attribute used for large block-aligned sequential I/O. The `setfa`(1) command's `-D` option is the direct I/O option. It sets the direct I/O attribute for a file or directory. If applied to a directory, the direct I/O attribute is inherited.

directory A file data structure that points to other files and directories within the file system.

disk allocation unit See DAU.

disk buffer When using StorageTek ASM-Remote software, the disk buffer is a buffer on the server system that is used when archiving data from the client to the server.

disk cache The disk-resident portion of the StorageTek ASM file system software. It is used to create and manage data files between online disk cache and archive media. Individual disk partitions or an entire disk can be used as disk cache.

disk space thresholds An administrator-defined amount of disk space that is available to a user. This defines the range of desirable disk cache utilization. The high threshold indicates the maximum level of disk cache utilization. The low threshold indicates the minimum level of disk cache utilization. The releaser controls disk cache utilization based on these predefined disk space thresholds.

disk striping The process of recording a file across several disks, thereby improving access performance and increasing overall storage capacity. See also entries for striping.

drive A mechanism for transferring data to and from a removable media volume.

E

Ethernet A local-area, packet-switched network technology. Originally designed for coaxial cable, it is now found running over shielded, twisted-pair cable. Ethernet is a 10- or 100-Mbytes/sec LAN.

extent array The array within a file's inode that defines where each data block assigned to the file is located on the disk.

F

family device set See Family Set.

Family Set A storage device that is represented by a group of independent physical devices, such as a collection of disks or the drives within an automated library. Also see storage Family Set.

FDDI (Fiber distributed data interface) A 100-Mbytes/sec fiber-optic LAN.

fibre channel The ANSI standard that specifies high-speed serial communication between devices. Fibre channel is used as one of the bus architectures in SCSI-3.

fiber-distributed data interface See FDDI.

file system A hierarchical collection of files and directories.

file system specific directives Archiver and releaser directives that follow global directives, are specific to a particular file system, and begin with `fs =`. File system specific directives apply until the next `fs =` directive line or until the end of file is encountered. If multiple directives affect a file system, the file system-specific directives override the global directives.

FTP (File transfer protocol) An internet protocol for transferring files between two hosts over a TCP/IP network.

G

global directives Archiver and releaser directives that apply to all file systems and that appear before the first `fs =` line.

grace period For disk quotas, this is the amount of time that can elapse during which a user is allowed to create files and allocate storage after the user reaches their soft limit.

H

hard limit For disk quotas, a maximum limit on file system resources, blocks and inodes, that users cannot exceed.

I

indirect block A disk block that contains a list of storage blocks. The StorageTek QFS and StorageTek ASM file systems have up to three levels of indirect blocks. A first-level indirect block contains a list of blocks used for data storage. A second-level indirect block contains a list of first-level indirect blocks. A third-level indirect block contains a list of second-level indirect blocks.

inode Index node. A data structure used by the file system to describe a file. An inode describes all the attributes associated with a file other than the name. The attributes include ownership, access, permission, size, and the file location on the disk system.

inode file A special file (`.inodes`) on the file system that contains the inode structures for all files resident in the file system. All StorageTek QFS and StorageTek ASM inodes are 512 bytes long. The inode file is a metadata file, which is separated from file data in the StorageTek QFS file systems.

K

kernel The central controlling program that provides basic system facilities. The UNIX kernel creates and manages processes, provides functions to access the file system, provides general security, and supplies communication facilities.

L

LAN Local area network.

lease In a StorageTek QFS shared file system, a lease grants a client host permission to perform an operation on a file for as long as the lease is valid. The metadata server issues leases to each client host. The leases are renewed as necessary to permit continued file operations.

library See automated library.

library catalog See catalog.

local file system A file system that is installed on one node of a Sun Cluster and is not made highly available to another node. Also a file system that is installed on a standalone server.

LUN Logical unit number.

M

mcf Master configuration file. The file that is read at initialization time that defines the relationships between the devices (the topology) within a StorageTek QFS or StorageTek ASM environment.

media Tape or optical disk cartridges.

media recycling The process of recycling or reusing archive media with low use. Archive media with low use is archive media with few active files.

metadata Data about data. Metadata is the index information needed to locate the exact data position of a file on a disk. It consists of information about files, directories, access control lists, symbolic links, removable media, segmented files, and the indexes of segmented files. Metadata must be protected because if data is lost, the metadata that locates the data must be restored before the lost data can be retrieved.

metadata device A separate device (for example, a solid-state disk or mirrored device) upon which StorageTek QFS file system metadata is stored. Separating the file data from the metadata can increase performance. In the `mcf` file, a metadata device is declared as an `mm` device within an `ma` file system.

mirror writing The process of maintaining two copies of a file on disjointed sets of disks to prevent loss from a single disk failure.

mount point The directory on which a file system is mounted.

multireader file system The StorageTek QFS multireader file system is a single-writer, multireader capability that enables you to specify a file system that can be mounted on multiple hosts. Multiple hosts can read the file system, but only one host can write to the file system. Multiple readers are specified with the `-o reader` option on the `mount`(1M) command. The single-writer host is specified with the `-o writer` option on the `mount`(1M) command. For more information on the `mount`(1M) command, see the `mount_samfs`(1M) man page.

N

name space The metadata portion of a collection of files that identifies the file, its attributes, and its storage locations.

nearline storage Removable media storage that requires robotic mounting before it can be accessed. Nearline storage is usually less expensive than online storage, but it incurs a somewhat longer access time.

network-attached automated library A library, such as those from StorageTek, ADIC/Grau, IBM, or Sony, that is controlled using a software package supplied by the vendor. The StorageTek ASM file system interfaces with the vendor software using a StorageTek ASM media changer daemon designed specifically for the automated library.

NFS Network file system. A Sun distributed file system that provides transparent access to remote file systems on heterogeneous networks.

NIS The SunOS 4.0 (minimum) Network Information Service. A distributed network database containing key information about the systems and the users on the network.

The NIS database is stored on the master server and all the slave servers.

O

offline storage Storage that requires operator intervention for loading.

offsite storage Storage that is remote from the server and is used for disaster recovery.

online storage Storage that is immediately available (for example, disk cache storage).

P

partition A portion of a device or a side of a magneto-optical cartridge.

preallocation The process of reserving a contiguous amount of space on the disk cache for writing a file. This ensures that the space is contiguous. Preallocation can be performed only on zero-sized files. That is, the `setfa -l` command can be specified only for a file that is size zero. For more information, see the `setfa`(1) man page.

prioritizing preview requests Assigning priority to archive and stage requests that cannot be immediately satisfied.

pseudo device A software subsystem or driver with no associated hardware.

Q

quota The amount of system resources that a user is allowed to consume.

R

RAID Redundant array of independent disks. A disk technology that uses several independent disks to reliably store files. It can protect against data loss from a single disk failure, can provide a fault-tolerant disk environment, and can provide higher throughput than individual disks.

recycler A StorageTek ASM utility that reclaims space on cartridges that is occupied by expired archive copies.

release priority A method of calculating the release priority of a file within a file system by multiplying various weights by the corresponding file properties and then summing the results.

releaser A StorageTek ASM component that identifies archived files and releases their disk cache copies, thus making more disk cache space available. The releaser automatically regulates the amount of online disk storage to high and low thresholds.

remote procedure calls See RPC.

removable media file A special type of user file that can be accessed directly from where it resides on a removable media cartridge, such as magnetic tape or optical disk cartridge. Also used for writing archive and stage file data.

robot The portion of an automated library that moves cartridges between storage slots and drives. Also called a transport.

round robin A data access method in which entire files are written to logical disks in a sequential fashion. When a single file is written to disk, the entire file is written to the first logical disk. The second file is written to the next logical disk, and so on. The size of each file determines the size of the I/O.

By default, StorageTek QFS and StorageTek ASM file systems implement striped data access unless striped groups are present. Files are round-robined if round robin access is specified. If the file system contains mismatched striped groups, striping is not supported and round robin is forced.

Also see glossary entries for disk striping and striping.

RPC Remote procedure calls. The underlying data exchange mechanism used by NFS to implement custom network data servers.

S

samfsdump A program that creates a control structure dump and copies all the control structure information for a given group of files. It is analogous to the UNIX `tar`(1) utility, but it does not generally copy file data.

samfsrestore A program that restores inode and directory information from a control structure dump.

SCSI Small Computer System Interface. An electrical communication specification commonly used for peripheral devices such as disk and tape drives and automated libraries.

small computer system interface See SCSI.

soft limit For disk quotas, a threshold limit on file system resources (blocks and inodes) that you can temporarily exceed. Exceeding the soft limit starts a timer. When you exceed the soft limit for the specified time (default is one week), no further system resources can be allocated until you reduce file system use to a level below the soft limit.

staging The process of copying a nearline or offline file from archive storage back to online storage.

storage Family Set A set of disks that are collectively represented by a single disk family device.

storage slots Locations inside an automated library in which cartridges are stored when not being used in a drive. If the library is direct-attached, the contents of the storage slots are kept in the automated library's catalog.

stripe size The number of disk allocation units (DAUs) to allocate before moving to the next device of a stripe. If `stripe=0`, the file system uses round-robin access, not striped access.

striped group A collection of devices within a StorageTek QFS file system and defined in the `mcf` file as one (usually two) or more `g`*XXX* devices. Striped groups are treated as one logical device and are always striped with a size equal to the disk allocation unit (DAU). You can specify up to 128 striped groups within a file system, but you can specify no more than 252 total devices.

striping A data access method in which files are simultaneously written to logical disks in an interlaced fashion. All StorageTek QFS and StorageTek ASM file systems enable you to declare either striped or round robin access for each individual file system. The StorageTek QFS file systems enable you to declare striped groups within each file system. Also see the glossary entry for round robin.

StorageTek QFS A high-speed UNIX file system that separates the file system metadata from the file data by storing them on separate devices. The StorageTek QFS software controls the access to all files stored and all devices configured in the master configuration file (`mcf`).

StorageTek ASM StorageTek Application Storage Manager. The StorageTek ASM software controls the access to all files stored and all devices configured in the master configuration file (`mcf`).

StorageTek ASM-QFS The StorageTek ASM-QFS software combines the StorageTek ASM software with the StorageTek QFS file system. StorageTek ASM-QFS offers a high-speed, standard UNIX file system interface to users and administrators in conjunction with the storage and archive management utilities. It uses many of the commands available in the StorageTek ASM command set as well as standard UNIX file system commands.

StorageTek ASM-Remote client A StorageTek ASM-Remote client is a StorageTek ASM system that establishes a StorageTek ASM-Remote client daemon that contains a number of pseudodevices. It might or might not have its own library devices. The client depends on a StorageTek ASM-Remote server for archive media for one or more archive copies.

StorageTek ASM-Remote server The StorageTek ASM-Remote server is both a full-capacity StorageTek ASM storage management server and a StorageTek ASM-Remote server daemon that defines libraries to be shared among StorageTek ASM-Remote clients.

superblock A data structure in the file system that defines the basic parameters of the file system. It is written to all partitions in the storage Family Set and identifies the partition's membership in the set.

T

tar Tape archive. A standard file and data recording format used by the StorageTek ASM software for archive images.

TCP/IP Transmission Control Protocol/ Internet Protocol. The internet protocols responsible for host-to-host addressing and routing, packet delivery (IP), and reliable delivery of data between application points (TCP).

thresholds A mechanism for defining the desirable available storage window for online storage. Thresholds set the storage goals for the releaser. Also see disk space thresholds.

timer Quota software that keeps track of the time elapsed between a user reaching a soft limit and a hard limit being imposed on the user.

V

volume A named area on a cartridge for sharing data. A cartridge has one or more volumes. Double-sided cartridges have two volumes, one on each side.

volume overflow A capability that enables the system to span a single file over multiple volumes. Volume overflow is useful for sites using very large files that exceed the capacity of their individual cartridges.

VSN Volume serial name. If you are archiving to removable media cartridges, the VSN is a logical identifier for magnetic tape and optical disk that is written in the volume label. If you are archiving to disk cache, this is the unique name for the disk archive set.

W

WORM Write once read many. A storage classification for media that can be written only once but read many times.

Index

Symbols

A

B

C

D

E

F

G

H

I

S

Z

NEED MORE INFORMATION?

www.storagetek.com

ABOUT STORAGETEK

Storage Technology Corporation (NYSE: STK) is a $2 billion global company that enables businesses, through its information lifecycle management strategy, to align the cost of storage with the value of information. The company's innovative storage solutions manage the complexity and growth of information, lower costs, improve efficiency and protect investments. For more information, visit www.storagetek.com, or call 1.800.275.4785 or 01.303.673.2800.

WORLD HEADQUARTERS

Storage Technology Corporation
One StorageTek Drive
Louisville, Colorado 80028 USA
1.800.678.4430 or 01.303.673.4430