Sun StorEdge™ QFS Installation and Upgrade Guide

Version 4, Update 4

Sun Microsystems, Inc.
www.sun.com

Part No. 819-2757-10
Sept. 2005, Revision A

Submit comments about this document at: `http://www.sun.com/hwdocs/feedback`

Contents

Preface xiii

How This Book Is Organized xiii

Using UNIX Commands xiv

Shell Prompts xiv

Typographic Conventions xv

Related Documentation xvi

Accessing Sun Documentation Online xvi

- To Access Documentation From `docs.sun.com` xvi
- To Access Documentation From Sun's Network Storage Documentation Web Site xvii

Third-Party Web Sites xvii

Contacting Sun Technical Support xvii

Licensing xviii

Installation Assistance xviii

Sun Welcomes Your Comments xviii

1. Planning Your File System 1

Product Overview 1

About the SAM-QFS File System 2

About Shared Sun StorEdge QFS File Systems and the Linux Client 2

About the Sun Cluster Environment 3
Installation Process Overview 4

2. Pre-installation Tasks 5

Hardware and Software Requirements 5
Hardware Requirements 5
Operating System Requirements 6
▼ To Verify the Environment 6
Installing Solaris OS Patches 6
Verifying Third-Party Compatibilities 7
Verifying Sun StorEdge QFS Shared File System Requirements 7
Metadata Server Requirement 7
Operating System and Hardware Requirements 7
Sun StorEdge QFS Release Levels 8
Verifying Requirements for a Sun Cluster Environment 8
Example: Verifying Devices and Device Redundancy 12
Performance Considerations 14
Verifying Requirements for File System Manager 15
Hardware Requirements 15
Browser Requirements 15
Operating System Requirements 16
Web Software Requirements 16
Determining Disk Space Requirements 16
Planning Your File System and Verifying Disk Cache 17
▼ To Estimate Disk Cache Requirements 17
Verifying Disk Space 18
▼ To Verify Disk Space 18
Obtaining the Release Files 19
▼ To Obtain the Software From the Sun Download Center 20

Software Licensing 20

Setting Up the Network Management Station 20

3. Installation and Configuration Tasks 23

Installing the Software Packages 24

- To Add the Packages 24

Installing the Linux Client Software 25

- To Set Up `PATH` and `MANPATH` Variables 25

Installing and Using the File System Manager Software 26

- To Install the File System Manager Software 26

Using the File System Manager Software 28

- To Invoke File System Manager for the First Time 29

Setting Up the Environment Configuration 30

- To Create an `mcf` File Using the File System Manager Software 31
- To Create an `mcf` File Using a Text Editor 32
- To Verify the `mcf` File 33

Setting Up Mount Parameters 34

Updating the `/etc/vfstab` File and Creating the Mount Point 35

- To Update the `/etc/vfstab` File Using File System Manager 36
- To Update the `/etc/vfstab` File and Create the Mount Point Using a Text Editor 36

Creating and Editing the `samfs.cmd` File 38

- To Create and Edit the `samfs.cmd` File Using File System Manager 39
- To Create and Edit the `samfs.cmd` File Using a Text Editor 39

Initializing the Environment 40

- To Initialize the Environment 40
- To Initialize the File System 41

Initializing a Sun StorEdge QFS File System 41

Initializing a Sun StorEdge QFS Shared File System 41

Mounting the File System 42

- To Mount the File System Using File System Manager 43
- To Mount the File System on One Host From the Command Line 43
- To Verify Metadata Server Changes 44

Performing Additional Configuration Tasks 45

Sharing the File System With NFS Client Systems 45

- To NFS-Share the File System in a Sun Cluster Environment 45
- To NFS-Share the File System in a Solaris OS Environment 46

Notes on NFS-Sharing 47

- To Mount the File System on NFS Clients in a Solaris OS Environment 47

- To Change Default Values 48

Configuring the Remote Notification Facility 49

- To Enable Remote Notification 50
- To Disable Remote Notification 51

Adding the Administrator Group 52

- To Add the Administrator Group 52

- To Enable Logging 52

Configuring Other Products 53

Backing Up Data 54

Setting Up Dump Files 54

- To Run the `qfsdump`(1M) Command Automatically Using `cron` 55
- To Run the `qfsdump`(1M) Command Manually From the Command Line 55

Backing Up Configuration Files 56

4. Configuration Tasks for a Shared or Sun Cluster Configuration 57

Preparing the Host Systems 57

- To Prepare the Host Systems 58

Editing `mcf` Files on Other Hosts 59

▼ To Edit `mcf` Files for a Highly Available File System in a Sun Cluster Environment 59

▼ To Edit `mcf` Files for a Sun StorEdge QFS Shared File System 60

Examples 61

Creating the Shared Hosts File 63

▼ To Create the Shared Hosts File on the Metadata Server 63

Example for Solaris OS Hosts 65

Example for Sun Cluster Hosts 65

▼ To Create the Local Hosts File on a Client 66

How Metadata Server Addresses Are Obtained 67

Example 67

Verifying That the Daemons Are Running 69

▼ To Verify the Daemons 69

Configuring the `SUNW.qfs` Resource Type 70

▼ To Enable a Sun StorEdge QFS Shared File System as a `SUNW.qfs`(5) Resource 71

Configuring the HA Storage Plus Resource 72

▼ To Configure a Highly Available File System as an HA Storage Plus Resource 72

Bringing the Shared Resource Online 72

▼ To Bring the Shared Resource Online 73

Verifying the Resource Group on All Nodes 73

▼ To Verify the Resource Group on All Nodes 74

5. Upgrade and Configuration Tasks 75

Preparing for an Upgrade 75

Upgrade Considerations 75

Preserving Information for an Upgrade 76

Example 1 77

Example 2 78

Preparing for a Hardware Device Upgrade 79
General Prerequisites 79
Switching Between SPARC and AMD Platforms 80
Backing Up Existing File Systems 83
Using the Version 1 and Version 2 Superblocks 83
▼ To Back Up a File System 84
▼ To Unshare File Systems 86
Unmounting File Systems 86
▼ To Unmount Using File System Manager 86
▼ To Unmount Using CLI Commands 86
▼ To Unmount by Editing the `/etc/vfstab` File and Rebooting 87
Removing Existing Sun StorEdge QFS Software 87
▼ To Remove Existing Software 87
Adding the Upgrade Packages 88
▼ To Add the Packages 88
Installing File System Manager 89
▼ To Install the File System Manager Software 89
Restoring the File System 91
▼ To Verify the `mcf` File 91
▼ To Modify the `/etc/vfstab` File 92
▼ To Reinitialize and Restore the File System 93
To Check the File System 94
Mounting the File System 94
▼ To Mount the File System Using File System Manager 94
▼ To Mount the File System Using the CLI 94
Recompiling API-Dependent Applications 94
Upgrading the Solaris OS 95
▼ To Upgrade the Solaris OS in a Sun StorEdge QFS Environment 95

A. Release Package Contents 97

Release Package Contents 97

Directories and Files Created 98

Directories Created at Installation 98

Files Created at Installation 99

Site Files 101

Modified System Files 101

B. Uninstalling the Software 103

Uninstalling the Sun StorEdge QFS Linux Client Software 103

Uninstalling the File System Manager Software 104

C. Command Reference 105

User Commands 105

General System Administrator Commands 106

File System Commands 107

Application Programming Interface 108

Operational Utilities 108

D. `mcf` File Examples 109

Configuration Examples for Local File Systems 109

Configuration Example 1 110

▼ To Configure the System 111

Configuration Example 2 112

▼ To Configure the System 112

Configuration Example 3 114

▼ To Configure the System 114

Configuration Example 4 115

▼ To Configure the System 116

Configuration Example for a Shared File System on a Solaris OS Platform 118

▼ To Configure the System 119

Configuration Examples for Highly Available File Systems 122

▼ To Create an `mcf` File for a Highly Available File System 122

Configuration Example for a Shared File System on a Sun Cluster Platform 124

▼ To Create an `mcf` File for a Shared File System in a Sun Cluster Environment 124

Glossary 129

Index 139

Tables

TABLE P-1 Shell Prompts xiv

TABLE P-2 Typographic Conventions xv

TABLE P-3 Related Sun StorEdge QFS Documentation xvi

TABLE 2-1 Minimum Disk Space Requirements 18

TABLE 3-1 Fields in the `/etc/vfstab` File 35

TABLE 3-2 Mount Options for a Sun Cluster File System 37

TABLE 3-3 `sammkfs`(1M) Command Arguments 42

TABLE 4-1 Hosts File Fields 63

TABLE 4-2 Local Hosts Configuration File Fields 66

TABLE A-1 Release Packages 97

TABLE A-2 Directories Created 98

TABLE A-3 Files Created - Miscellaneous 99

TABLE A-4 Files Created - Fault Notification 100

TABLE C-1 User Commands 105

TABLE C-2 General System Administrator Commands 106

TABLE C-3 File System Commands 107

TABLE C-4 Operational Utilities 108

Preface

This manual explains the installation and upgrade procedures for the Sun StorEdge QFS software product, Version 4, Update 4 (4U4). The 4U4 release can be installed on the following minimum Operating System (OS) platforms:

- Solaris 9 OS 04/03 on SPARC platforms
- Solaris 10 OS on SPARC or x64 platforms
- Red Hat 3 Linux, and SuSE 8 Linux (shared clients only)

Certain features might require a specific operating system level. For more information, see the release notes or see the specific feature's documentation.

This manual is written for system administrators responsible for configuring and maintaining Sun StorEdge QFS software. You, the system administrator, are assumed to be knowledgeable about Sun Solaris procedures, including creating accounts, performing system backups, creating file systems, and other basic Sun Solaris system administrator tasks.

How This Book Is Organized

This manual contains the following chapters:

- Chapter 1 has information about planning your file system.
- Chapter 2 describes system requirements and pre-installation tasks.
- Chapter 3 explains the Sun StorEdge QFS initial installation procedure.
- Chapter 4 provides some additional installation instructions for Sun StorEdge QFS shared and Sun Cluster environments.
- Chapter 5 explains the Sun StorEdge QFS upgrade procedure.

- Appendix A describes the release package contents and the directories created at installation time.
- Appendix B gives instructions for uninstalling the Sun StorEdge QFS and File System Manager software.
- Appendix C is a command reference.
- Appendix D provides configuration (`mcf`) file examples.

The glossary defines terms used in this and other Sun StorEdge QFS and Sun StorEdge SAM-FS documentation.

Using UNIX Commands

This document does not contain information about 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 P-1 shows the shell prompts used in this manual.

TABLE P-1 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 P-2 lists the typographic conventions used in this manual.

TABLE P-2 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`

Related Documentation

This manual is part of a set of documents that describes the operations of the Sun StorEdge QFS and Sun StorEdge SAM-FS software products. TABLE P-3 shows the complete release 4U4 documentation set for these products.

TABLE P-3 Related Sun StorEdge QFS Documentation

Title	Part Number
Sun StorEdge QFS Configuration and Administration Guide	819-2758-10
Sun StorEdge SAM-FS File System Configuration and Administration Guide	819-3635-10
Sun StorEdge SAM-FS Storage and Archive Management Guide	819-2755-10
Sun StorEdge SAM-FS Installation and Upgrade Guide	819-2754-10
Sun StorEdge SAM-FS Troubleshooting Guide	819-2756-10
Sun StorEdge QFS and Sun StorEdge SAM-FS 4.4 Release Notes	819-2759-10

If you are configuring a Sun StorEdge QFS file system in a Sun Cluster environment, the following additional documents might interest you:

- *Sun Cluster Concepts Guide for Solaris OS*
- *Sun Cluster Software Installation Guide for Solaris OS*
- *Sun Cluster Data Services Planning and Administration Guide for Solaris OS*
- *Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS*

Accessing Sun Documentation Online

The Sun StorEdge QFS software distribution includes PDF files that you can view from Sun's Network Storage documentation web site or from `docs.sun.com`.

▼ To Access Documentation From `docs.sun.com`

This web site contains documentation for Solaris and many other Sun software products.

1. **Go to the following URL:**

```
http://docs.sun.com
```

The `docs.sun.com` page appears.

2. **Find the documentation for your product by searching for Sun StorEdge QFS in the search box.**

▼ To Access Documentation From Sun's Network Storage Documentation Web Site

This web site contains documentation for Network Storage products.

1. **Go to the following URL:**

```
http://www.sun.com/products-n-solutions/hardware/docs/Software
/Storage_Software
```

The Storage Software page appears.

2. **Click the Sun StorEdge QFS Software link.**

Third-Party Web Sites

Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Contacting Sun Technical Support

If you have technical questions about this product that are not answered in this document, go to the following web site:

```
http://www.sun.com/service/contacting
```

Licensing

For information about obtaining licenses for Sun StorEdge QFS software, contact your Sun sales representative or your authorized service provider (ASP).

Installation Assistance

For installation and configuration services, contact Sun's Enterprise Services at 1-800-USA4SUN or contact your local Enterprise Services sales representative.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can submit your comments by going to the following web site:

`http://www.sun.com/hwdocs/feedback`

Please include the title and part number of your document with your feedback (*Sun StorEdge QFS Installation and Upgrade Guide*, part number 819-2757-10).

CHAPTER 1

Planning Your File System

This chapter provides an overview of the Sun StorEdge QFS software and explains some of the design considerations that should be taken into account before you begin to install and configure your software. It also provides a high level overview of the software installation process.

This chapter contains the following sections:

- "Product Overview" on page 1
- "Installation Process Overview" on page 4

Product Overview

Sun StorEdge QFS software is a high-performance file system that can be installed on Solaris x64 AMD and SPARC platforms, and on Linux x86/x64 platforms (for shared clients only). This high-availability file system ensures that data is available at device-rated speeds when requested by one or more users. The Sun StorEdge QFS file system's inherent scalability enables the storage requirements of an organization to grow over time with virtually no limit to the amount of information that can be managed. This file system enables you to store many types of files (text, image, audio, video, and mixed-media) all in one logical place. In addition, the Sun StorEdge QFS file system enables you to implement disk quotas and a shared file system. This file system also includes the following features:

- Metadata separation
- Direct I/O capability
- Shared reader/writer capability
- Sun Cluster support for high availability
- File sharing in a storage area network (SAN) environment

About the SAM-QFS File System

The SAM-QFS configuration is one in which the Sun StorEdge QFS file system is used with the Sun StorEdge SAM-FS storage and archive management software. The SAM-QFS file system enables data to be archived to and retrieved from automated libraries at device-rated speeds. This file system manages the online, nearline, and offline data automatically and in a manner that is transparent to the user or application. Users read and write files to a SAM-QFS file system as though all files were on primary storage. In addition, the SAM-QFS file system backs up work-in-progress continually, automatically, and unobtrusively. Multiple file copies can be archived to many media types in a standard format. This minimizes the need for additional backup and provides fast disaster recovery in an effective long-term data storage solution.

The SAM-QFS file system is especially suited to data-intensive applications that require a scalable and flexible storage solution, superior data protection, and fast disaster recovery. This file system also includes the following features:

- Storage policy management
- Complete volume manager
- Disk-to-disk copying and archiving
- Shared tape drives
- Read-behind capability
- File segmentation

For more information on the Sun StorEdge SAM-FS product, see the *Sun StorEdge SAM-FS File System Configuration and Administration Guide*, the *Sun StorEdge SAM-FS Installation and Upgrade Guide*, and the *Sun StorEdge SAM-FS Storage and Archive Management Guide*.

About Shared Sun StorEdge QFS File Systems and the Linux Client

A Sun StorEdge QFS shared file system is a distributed, multihost file system that you can mount on multiple Solaris Operating System (OS) hosts. One Solaris OS host acts as the metadata server, and the others act as clients. You can also designate one or more clients as potential metadata servers, enabling you to switch metadata servers. FIGURE 1-1 illustrates the configuration of a simple Sun StorEdge QFS shared file system.

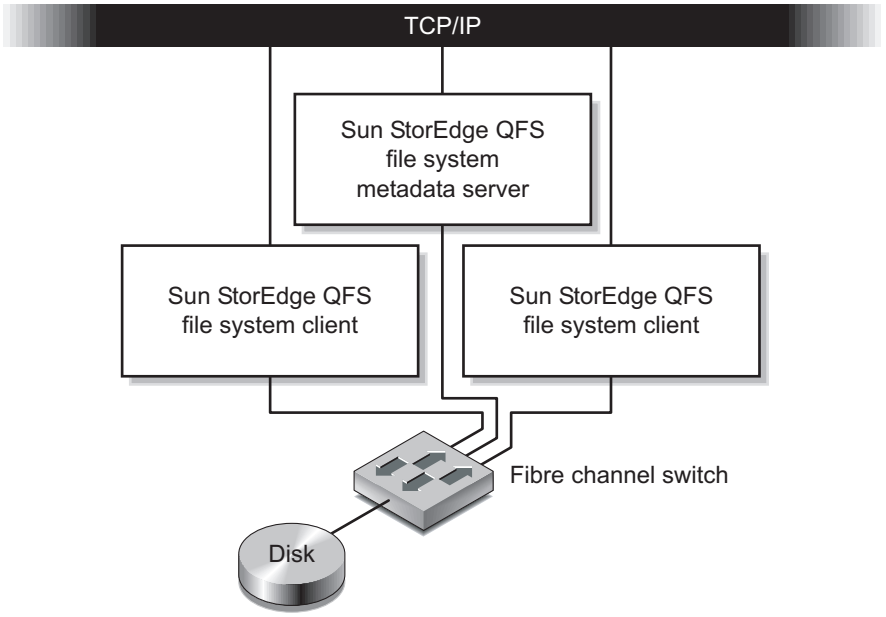

FIGURE 1-1 Sun StorEdge QFS Shared File System Configuration on Solaris Hosts

Within a shared file system, the Sun StorEdge QFS software can be installed on Linux clients as well as on Solaris clients. The Sun StorEdge QFS Linux client software supports the following Linux distributions:

- Red Hat Enterprise 3.0, AS, ES, and WS for x86/x64 platforms
- SuSE Enterprise Server 8 for x64 platforms

Unlike a shared Sun StorEdge QFS Solaris client, the Linux client is restricted to client-only behavior. It cannot be configured as a potential metadata server. The Linux client supports interaction with Sun StorEdge SAM-FS software, but has Sun StorEdge QFS file system functionality only.

The Sun StorEdge QFS software functionality is largely the same for the Solaris and Linux clients. For more information about the Sun StorEdge QFS Linux client software, see the *Sun StorEdge QFS Configuration and Administration Guide* and the README file on Disk 1 of the Sun StorEdge QFS Linux Client installation package.

About the Sun Cluster Environment

You can install a Sun StorEdge QFS file system in a Sun Cluster environment and configure the file system for high availability. The following configuration methods are available, depending on whether your file system is shared or unshared:

- In a shared file system, when the Sun Cluster software fails over, it moves the Sun StorEdge QFS file system operations from the failing server to a different server. The Sun Cluster software moves the metadata server's operations from a failing node to another node without requiring you to enter commands to move the metadata server to another host.

  Make sure that your environment meets the requirements listed in "Hardware and Software Requirements" on page 5.

- An unshared Sun StorEdge QFS file system configured in a Sun Cluster environment is a highly available file system. Such a file system is configured on one node but is enabled as a highly available resource within the cluster. When the node hosting the file system fails, the Sun Cluster software moves the file system to another node.

Note – Sun StorEdge QFS version 4U4 does not support a Sun Cluster environment on an AMD x64 hardware platform.

Installation Process Overview

The following list is a high-level overview of the software installation process. For detailed installation instructions, see Chapter 3 of this manual.

1. Verify that the hardware and software requirements have been met (see "Hardware and Software Requirements" on page 5).
2. Install the software packages ("Installing the Software Packages" on page 24).
3. If you want to configure the system using File System Manager, install the File System Manager software ("Installing and Using the File System Manager Software" on page 26).
4. Configure the Sun StorEdge QFS environment (beginning with "Setting Up the Environment Configuration" on page 30).
5. If applicable to your environment, complete the configuration tasks specific to a shared Sun StorEdge QFS or Sun Cluster environment ("Configuration Tasks for a Shared or Sun Cluster Configuration" on page 57).

CHAPTER 2

Pre-installation Tasks

This chapter explains the system requirements for the Sun StorEdge QFS software and the tasks you must complete before you begin to install and configure your software.

This chapter contains the following sections:

- "Hardware and Software Requirements" on page 5
- "Determining Disk Space Requirements" on page 16
- "Obtaining the Release Files" on page 19
- "Setting Up the Network Management Station" on page 20

Hardware and Software Requirements

This section outlines the hardware and software requirements for the Sun StorEdge QFS software.

Hardware Requirements

The Sun StorEdge QFS software can be installed either on a Sun server based on UltraSPARC® technology or on a server based on AMD Opteron x64 technology.

If you plan to install the File System Manager browser interface tool, there are additional requirements for the server that you want to use as the web server host. For more information about these requirements, see "Verifying Requirements for File System Manager" on page 15.

Operating System Requirements

Sun StorEdge QFS software package runs in the following operating system environments:

- Solaris 9 OS 04/03
- Solaris 10 OS
- Red Hat Enterprise 3.0, AS, ES, and WS for x86/x64 platforms – shared client only
- SuSE Enterprise Server 8 for x64 platforms – shared client only

Before installation, you should verify the applicability of the hardware, the level of the operating system, and the patch release installed. To install the Sun StorEdge QFS software, you also must ensure that you have root-level access to your system.

▼ To Verify the Environment

Repeat these steps for each host on which you want to install the Sun StorEdge QFS software.

1. **Verify that your system has a CD-ROM drive or that it can access the release package at the Sun Download Center.**

   The Sun Download Center is at the following URL:

   `http://www.sun.com/software/downloads`

2. **Log in to your system as root.**

   You must have superuser access to install the software.

3. **Verify your system's Solaris OS level.**

   The software relies on properly configured Solaris software at one of the following minimum release levels:

   - Solaris 9 OS 04/03
   - Solaris 10 OS (for SPARC or x64 platforms)

Installing Solaris OS Patches

Sun Microsystems provides Solaris OS patches to customers with a maintenance contract by means of CD-ROM, anonymous FTP, and the Sun Microsystems SunSolve web site (`http://sunsolve.sun.com`).

To install a patch after you install the Sun StorEdge QFS release packages, load the CD-ROM or transfer the patch software to your system. Follow the instructions outlined in the *Patch Installation Instructions and Special Install Instructions* in the README file included in the patch or jumbo patch cluster.

Verifying Third-Party Compatibilities

The Sun StorEdge QFS software interoperates with many different hardware and software products from third-party vendors. Depending on your environment, you might need to upgrade other software or firmware before installing or upgrading the Sun StorEdge QFS package. Consult the *Sun StorEdge QFS and Sun StorEdge SAM-FS 4.4 Release Notes* for information pertaining to library model numbers, firmware levels, and other compatibility information.

Verifying Sun StorEdge QFS Shared File System Requirements

This section describes the system requirements for a Sun StorEdge QFS shared file system.

Metadata Server Requirement

There must be at least one Solaris metadata server. If you want to be able to change the metadata server, there must be at least one other Solaris host that can become the metadata server; these other host systems are known as potential metadata servers. These servers must all be running on the same hardware platform, either SPARC or x64. You cannot mix server hardware platforms. In a Sun Cluster environment, all nodes included in a Sun StorEdge QFS shared file system are potential metadata servers.

The following are configuration recommendations with regard to metadata storage:

- A Sun StorEdge QFS shared file system should have multiple metadata (`mm`) partitions. This spreads out metadata I/O and improves file system throughput.
- A Sun StorEdge QFS shared file system should use a separate, private metadata network so that typical user traffic does not interfere with metadata traffic. A switch-based (not hub-based) network is recommended for this.

Operating System and Hardware Requirements

Ensure that your configuration meets the following operating system and hardware requirements:

- The host systems to be configured in the Sun StorEdge QFS shared file system must be connected by a network.
- All metadata servers and potential metadata servers must have the same processor type.

- The client systems can be installed on the Solaris OS or on one of the following OSs:
  - Red Hat Enterprise 3.0, AS, ES, and WS for x86/x64 platforms
  - SuSE Enterprise Server 8 for x64 platforms
- Online data storage devices must be directly accessible to all hosts. All online metadata storage devices must be directly accessible to all potential metadata server hosts.

Sun StorEdge QFS Release Levels

Ensure that your configuration meets the following Sun StorEdge QFS requirements:

- Each host to be configured in the Sun StorEdge QFS shared file system must have a Sun StorEdge QFS software package installed upon it.
- All Sun StorEdge QFS software installed on the systems in the Sun StorEdge QFS shared file system must be at the same release level. This ensures that all systems in a Sun StorEdge QFS shared file system have identical over-the-wire protocol versions. If these levels do not match, the system generates the following message when mounting is attempted:

```
SAM-FS: client client package version x mismatch, should be y.
```

  The system writes the preceding message to the metadata server's `/var/adm/messages` file.
- When applying patches or upgrading the software for a Sun StorEdge QFS shared file system, make sure to apply the same patch to all hosts that have access to the shared file system. Unexpected results might occur if not all host systems are running the same patch level.

Verifying Requirements for a Sun Cluster Environment

If you plan to configure a Sun StorEdge QFS file system in a Sun Cluster environment, verify the following:

1. **Verify your hardware.**

   Ensure that you have between two and eight UltraSPARC hosts to use as a cluster.

Note – Sun StorEdge QFS version 4U4 does not support a Sun Cluster environment on an AMD x64 hardware platform.

2. **Verify your software.**

   Ensure that you have the following minimum software levels installed on each cluster node:

   - Solaris 9 OS 04/03 or Solaris 10 OS
   - Sun Cluster 3.1 4/04

   Each node must have the same Sun Cluster software level and Sun Cluster patch collection. You must install Sun StorEdge QFS software packages on each node in the cluster that will host a Sun StorEdge QFS file system.

3. **Ensure that you are familiar with how disks are used in a Sun Cluster environment.**

   In a Sun Cluster environment, the disk space used by the Sun StorEdge QFS file system must be configured on storage that is highly available and redundant. Ensure that you have a good understanding of the concepts in the *Sun Cluster System Administration Guide for Solaris OS*.

   You should also be familiar with Sun Cluster operations. For information on Sun Cluster operations, see the following manuals:

   - *Sun Cluster Concepts Guide for Solaris OS*
   - *Sun Cluster Software Installation Guide for Solaris OS*
   - *Sun Cluster Data Services Planning and Administration Guide for Solaris OS*

4. **Verify your disk space.**

   "Verifying Disk Space" on page 18 explains how much disk space to allow for the various directories that the file systems need.

5. **Verify that you have the correct kinds of disk devices.**

   The types of disk devices you can use depend on the kind of file system you are configuring and whether you are using a volume manager, as follows:

   - If you are going to configure a Sun StorEdge QFS shared file system, use raw device identification (DID) devices. In the `scdidadm`(1M) command output, these appear as `/dev/did/*` devices. The Sun Cluster nodes that share the file system must have access to each DID device through a host bus adapter (HBA) direct connection. All devices must be accessible to the file system from all nodes in the Sun Cluster environment that mount the Sun StorEdge QFS shared file system. For more information about DID devices, see the `did`(7) man page.

     When you specify these devices in your `mcf` file, use the `/dev/did` devices from the `scdidadm`(1M) output. For more information about using `scdidadm`(1M), see "Example: Verifying Devices and Device Redundancy" on page 12.

Note – Starting with version 4U4 the Sun StorEdge QFS software supports using multi-owner disksets in Solaris Volume Manager for Sun Cluster to obtain redundancy. In versions previous to 4U4, you should not use a volume manager with a Sun StorEdge QFS shared file system in a Sun Cluster environment. Data corruption can result.

- If you are going to configure a Sun StorEdge QFS highly available file system, you must use highly available devices. You can use either raw devices or devices managed by a volume manager.

  If you want to configure from raw devices, use Sun Cluster global devices. Use the output from the `scdidadm`(1M) command to determine the names of the global devices and substitute `global` for `did` when specifying the devices in the `mcf`(1) file. Global devices are accessible from all nodes in a Sun Cluster environment, even if these devices are not physically attached to all nodes. If all nodes that have a hardware connection to the disk lose their connections, the remaining nodes cannot access the disk. File systems created on global devices are not necessarily highly available.

  If you want to use a volume manager, use one of the following:

  - Solstice DiskSuite volume manager. Such devices are located in `/dev/md`.
  - VERITAS Volume Manager (VxVM). Such devices are located in `/dev/vx`.

  Use `scsetup`(1M) to register volume-managed devices with the Sun Cluster framework prior to configuring your file system.

Note – If you use a volume manager, use it only to provide redundancy. For performance reasons, do not use it to concatenate storage on separate devices; this causes the Sun StorEdge QFS highly available file system to distribute I/O inefficiently across the component devices.

If you are unsure about your devices, issue the `scdidadm`(1M) command with its `-L` option to determine which devices in your Sun Cluster environment are highly available. This command lists the paths of the devices in the DID configuration file. In the output from the `scdidadm`(1M) command, look for devices that have two or more DID devices listed with the exact same DID device number. Such devices are highly available in a Sun Cluster environment and can also be configured as global devices for a file system, even if they directly connect only to a single node.

I/O requests issued to global devices from a node other than the direct attached node are issued over the Sun Cluster interconnect. These single-node, global devices cease to be available when all nodes that have direct access to the device are unavailable.

6. **Verify device redundancy.**

   There are two types of redundancy to consider in a Sun Cluster environment: storage redundancy and data path redundancy. The implications of these are as follows:

   - Storage redundancy is achieved by maintenance of extra disk copies of data using mirroring or RAID-1, or parity across several disks using RAID-5 to allow reconstruction of data after a disk failure. When supported by the hardware, these disk configurations enable you to configure the raw devices in a Sun Cluster environment without a volume manager. These raw devices are accessible from multiple nodes, so you can issue the `format`(1M) command from any node to obtain information on the disks.

     Storage redundancy can also be achieved by using software to support mirroring or RAID. This method, however, is not generally suitable for concurrent access from multiple hosts. Sun Cluster software supports mirroring of disk volumes (RAID-1 only) through its multi-owner diskset feature with Sun StorEdge QFS software and Solaris Volume Manager. This requires Sun Cluster software version 3.1 8/05 or later, Sun StorEdge QFS software version 4U4 or later, and the Solaris Volume Manager patch for the Solaris 10 OS, which is pending release. No other software redundancy is supported.

   - Data path redundancy is achieved with multiple HBAs, which are configured from a single node. If your environment includes multiple HBAs for redundancy, be aware that the Sun StorEdge QFS file systems require multipathing software like Sun StorEdge Traffic Manager software (MPxIO) to enable data path redundancy. For more information, see the *Sun StorEdge Traffic Manager Software Installation and Configuration Guide*, or see the `scsi_vhci`(7D) man page.

   To determine redundancy, consult the hardware documentation for your disk controllers and disk devices. You need to know whether the disk controller or disk devices that are reported by `scdidadm`(1M) are on redundant storage. For information, see the storage controller vendor's documentation set and view the current controller configuration.

   After the set of highly available devices has been determined, check for device redundancy. All devices must employ mirroring (RAID-1) or striping (RAID-5) to ensure continued operation in the event of a failure, as follows:

   - If you are configuring a Sun StorEdge QFS shared file system, you have the option of obtaining redundancy through multi-owner disksets in Solaris Volume Manager for the Sun Cluster environment. This support was added in version 4U4 of the Sun StorEdge QFS software. If you are using an earlier version of the software, the redundancy must be supported in the disk device hardware. Do not use a volume manager to obtain redundancy.

- If you are configuring a Sun StorEdge QFS highly available file system, you can use either the Solstice DiskSuite volume manager or the VERITAS Volume Manager to obtain mirroring (RAID-1) or striping (RAID-5).

For more information about volume sizing and redundancy configurations, see the *Solaris Volume Manager Administration Guide* or your VERITAS Volume Manager documentation.

Example: Verifying Devices and Device Redundancy

This example shows how to use output from the `scdidadm`(1M) command to find the devices in the Sun Cluster environment, determine which devices are highly available, and then determine which devices are redundant.

Determining High Availability

CODE EXAMPLE 2-1 shows the `scdidadm`(1M) Sun Cluster command. This example uses the `-L` option for this command to list paths of the devices in the DID configuration file for all nodes. In the output from the `scdidadm`(1M) command, look for output that shows a device that is visible from two or more nodes and that has the same World Wide Name. These are global devices.

CODE EXAMPLE 2-1 uses Sun StorEdge T3 arrays in a RAID-5 configuration. The output shows that you can use devices 4 through 9 for configuring the disk cache for a file system.

CODE EXAMPLE 2-1 `scdidadm`(1M) Command Example

```
ash# scdidadm -L
1        ash:/dev/rdsk/c0t6d0        /dev/did/rdsk/d1
2        ash:/dev/rdsk/c1t1d0        /dev/did/rdsk/d2
3        ash:/dev/rdsk/c1t0d0        /dev/did/rdsk/d3
4        elm:/dev/rdsk/c6t50020F2300004921d1 /dev/did/rdsk/d4
4        ash:/dev/rdsk/c5t50020F2300004921d1 /dev/did/rdsk/d4
5        elm:/dev/rdsk/c6t50020F2300004921d0 /dev/did/rdsk/d5
5        ash:/dev/rdsk/c5t50020F2300004921d0 /dev/did/rdsk/d5
6        elm:/dev/rdsk/c6t50020F23000049CBd1 /dev/did/rdsk/d6
6        ash:/dev/rdsk/c5t50020F23000049CBd1 /dev/did/rdsk/d6
7        elm:/dev/rdsk/c6t50020F23000049CBd0 /dev/did/rdsk/d7
7        ash:/dev/rdsk/c5t50020F23000049CBd0 /dev/did/rdsk/d7
8        elm:/dev/rdsk/c6t50020F23000055A8d0 /dev/did/rdsk/d8
8        ash:/dev/rdsk/c5t50020F23000055A8d0 /dev/did/rdsk/d8
9        elm:/dev/rdsk/c6t50020F23000078F1d0 /dev/did/rdsk/d9
9        ash:/dev/rdsk/c5t50020F23000078F1d0 /dev/did/rdsk/d9
10       elm:/dev/rdsk/c0t6d0        /dev/did/rdsk/d10
11       elm:/dev/rdsk/c1t1d0        /dev/did/rdsk/d11
12       elm:/dev/rdsk/c1t0d0        /dev/did/rdsk/d12
```

CODE EXAMPLE 2-1 scdidadm(1M) Command Example *(Continued)*

```
      # The preceding output indicates that both ash and elm can access disks 4, 5, 6, 7, 8, and 9.
      # These disks are highly available.


ash# format /dev/did/rdsk/d5s2
selecting /dev/did/rdsk/d5s2
[disk formatted]


FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        repair     - repair a defective sector
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
        inquiry    - show vendor, product and revision
        volname    - set 8-character volume name
        !<cmd>     - execute <cmd>, then return
        quit
format> verify

Primary label contents:

Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 192 sec 64>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =  192
nsect       =   64
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264      101.16GB    (17265/0/0) 212152320
  1        usr    wm   17265 - 34529      101.16GB    (17265/0/0) 212152320
  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
```

Analyzing the Output From the Commands

The `scdidadm`(1M) command in this example lists device `/dev/rdsk/c6t50020F2300004921d0`, which is DID device `/dev/did/rdsk/d5` or global device `/dev/global/rdsk/d5`. This device has a two partitions (`0` and `1`), each of which yields 212152320 blocks for use by a Sun StorEdge QFS highly available file system as `/dev/global/rdsk/d5s0` and `/dev/global/rdsk/d5s1`.

You need to issue the `scdidadm`(1M) and `format`(1M) commands for all devices to be configured for use by the Sun StorEdge QFS highly available file system.

- If you want to configure a Sun StorEdge QFS shared file system on a cluster, you must use highly available, redundant devices.
- If you want to configure a Sun StorEdge QFS highly available file system and the `scdidadm`(1M) command output indicates that the devices you want to use are JBOD (just a bunch of disks) or dual-port SCSI disk devices, you need to use a volume manager that is supported in a Sun Cluster environment to obtain redundancy. The options available and capabilities provided by such a volume manager are beyond the scope of this manual.

  Version 4U4 of the software added support for multi-owner disksets in Solaris Volume Manager for the Sun Cluster environment to obtain redundancy with shared Sun StorEdge QFS file systems. If you are using an earlier version of the software, you cannot use a volume manager to construct redundant devices to support a Sun StorEdge QFS shared file system.

For more information about configuring devices that are on redundant storage, see your Sun Cluster software installation documentation.

Performance Considerations

For optimal file system performance, the metadata and file data should be accessible through multiple interconnects and multiple disk controllers. In addition, plan to write file data to separate, redundant, highly available disk devices.

Plan to write your file system's metadata to RAID-1 disks. You can write file data to either RAID-1 or RAID-5 disks.

If are configuring a Sun StorEdge QFS highly available file system and you are using a volume manager, the best performance is realized when the file system is striping data over all controllers and disks, rather than when the volume manager performs the striping. You should use a volume manager only to provide redundancy.

Verifying Requirements for File System Manager

Perform this verification if you want to use the File System Manager browser interface to configure, control, monitor, or reconfigure a Sun StorEdge QFS environment through a web server.

You can install the File System Manager software in one of the following configurations:

- As a standalone management station to manage one or more Sun StorEdge QFS hosts.
- As additional software on the Sun StorEdge QFS host.

After the File System Manager software is installed, you can invoke File System Manager from any machine on the network that is allowed access to its web server.

If you plan to use File System Manager, the host upon which you are configuring the File System Manager software must meet the requirements described in the following subsections:

- "Hardware Requirements" on page 15
- "Browser Requirements" on page 15
- "Operating System Requirements" on page 16
- "Web Software Requirements" on page 16

Hardware Requirements

The minimum hardware requirements for the File System Manager software are as follows:

- SPARC 400 MHz (or more) CPU or x64 AMD CPU
- 1 gigabyte of memory
- One 20-gigabyte disk
- One 10/100/1000Base-T Ethernet port

Browser Requirements

Ensure that your installation meets the following browser requirements:

- One of the following browsers, at the minimum level indicated, must be installed on any client system that needs to access File System Manager:
  - Netscape 7.*x* / Mozilla 1.2.1 on the Solaris OS or Microsoft Windows 98, SE, ME, 2000 or XP operating system

- Internet Explorer 5.5 on Microsoft Windows 98, SE, ME, 2000, or XP operating system

- You must enable JavaScript technology in your browser. In Mozilla, for example, click through the following menus to get to a panel showing whether JavaScript technology is enabled: `Edit`, `Preferences`, `Advanced`, and `Scripts & Plugins`.

Operating System Requirements

Make sure that one of the following minimum levels of the Solaris OS is installed on the web server:

- Solaris 9 OS 4/03
- Solaris 10 OS

Web Software Requirements

The File System Manager installation packages include revisions of the following software at the minimum levels indicated:

- Java 2 Standard Edition version 1.4.2
- JavaHelp 2.0
- JATO 2.1.2 or greater
- TomCat version 4.0.5

During the installation procedure, you will be asked to answer questions about what you currently have installed. Based on your answers, the installation software can install the correct revisions for you if the compatible revisions of these software packages are not present.

Determining Disk Space Requirements

This section describes how to estimate the size of disk cache needed for creation and management of files and directories in your file system.

Planning Your File System and Verifying Disk Cache

The Sun StorEdge QFS software requires a certain amount of disk cache (file system devices) in order for it to create and manage data files and directories. An `ma`-type file system requires at least two disk devices or partitions, one for file data and one for metadata. An `ms`-type file system only requires one partition, on which both data and metadata is saved. Multiple disk devices or partitions increase I/O performance. See the *Sun StorEdge QFS Configuration and Administration Guide* for a detailed description of the two file system types.

The disk devices or partitions do not require any special formatting. You might see better performance if you configure multiple devices across multiple interfaces (HBAs) and disk controllers.

Caution – Make sure that the disks and partitions that you plan to use are not currently in use and do not contain any existing data. Any existing data will be lost when you create the Sun StorEdge QFS file system.

The disks must be connected to the server through a Fibre Channel or SCSI controller. You can specify individual disk partitions for a disk, or you can use the entire disk as a disk cache. The software supports disk arrays, including those under the control of volume management software, such as Solstice DiskSuite, Solaris Volume Manager, and other volume management software products.

Before creating your first file system, you should familiarize yourself with Sun StorEdge QFS file system layout possibilities. Describing all the aspects of Sun StorEdge QFS file systems is beyond the scope of this manual. For information on volume management, file system layout, and other aspects of file system design, see the *Sun StorEdge QFS Configuration and Administration Guide*.

▼ To Estimate Disk Cache Requirements

1. **Estimate the minimum disk cache requirements for Sun StorEdge QFS software (`ma` file systems).**
   - Disk cache = largest file (in bytes) + amount of space needed for working files
   - Metadata cache

     Use the following information to estimate the metadata cache requirements. The metadata cache must have enough space to contain the following data:

     - Two copies of the superblock (16 Kbytes each)
     - Reservation maps for metadata space plus data space ((metadata + file data)/DAU/32,000) * 4 Kbytes

- Inode space
  (number of files + number of directories) * 512 bytes
- Indirect blocks – a minimum of 16 Kbytes each
- Directory data space
  (number of directories * 16 Kbytes)

2. **Enter the `format`(1M) command to verify that you have sufficient disk cache space.**

   Use the `format`(1M) command if you are installing a Sun StorEdge QFS file system on a single server or if you are installing a Sun StorEdge QFS file system as a local file system on a Sun Cluster node.

   The `format`(1M) command shows how the disks are partitioned and the size of each partition.

Verifying Disk Space

The software requires a disk cache consisting of RAID devices, a JBOD (just a bunch of disks) collection, or both. It also requires a certain amount of disk space in the `/` (root), `/opt`, and `/var` directories. The actual amount needed varies depending on the packages you install. TABLE 2-1 shows the minimum amount of disk space required in these various directories.

TABLE 2-1 Minimum Disk Space Requirements

Directory	Sun StorEdge QFS Minimums	File System Manager Minimum
`/`(root) directory	2 Mbytes	25 Mbytes
`/opt` directory	8 Mbytes	5 Mbytes
`/var` directory	1 Mbyte	2 Mbytes
`/usr` directory	2 Mbytes	7 Mbytes
`/tmp` directory	0 Mbytes	200 Mbytes

Note – To see minimum disk space requirements for the Sun StorEdge SAM-FS Software, see the *Sun StorEdge SAM-FS Installation and Upgrade Guide*.

▼ To Verify Disk Space

The following procedure shows how to verify whether there is enough disk space on your system to accommodate the `SUNWsamfsu` and `SUNWsamfsr` software installation packages.

1. **Enter the following command to verify that there are at least 2 Mbytes available in the `avail` column for the `/` directory.**

```
# df -k /
Filesystem        kbytes  used    avail capacity   Mounted on
/dev/dsk/c0t1dos0 76767   19826   49271    29%     /
```

2. **Enter the following command to verify that there are at least 8 Mbytes in the `avail` column for the `/opt` directory.**

```
# df -k /opt
Filesystem        kbytes  used    avail capacity   Mounted on
/dev/dsk/c0t1dos4 192423  59006  114177    35%     /opt
```

3. **Verify that there is at least 1 Mbyte available in the `/var` directory.**

   A quantity of 30 Mbytes or more is recommended to allow for the growth of log files and other system files.

4. **If there is not enough room for the software under each directory, repartition the disk to make more space available to each file system.**

   To repartition a disk, see your Sun Solaris system administration documentation.

Obtaining the Release Files

Make sure that you have a copy of the release software. You can obtain the Sun StorEdge QFS software from the Sun Download Center or on a CD-ROM. Contact your authorized service provider (ASP) or your Sun sales representative if you have questions on obtaining the software.

After the release, upgrade patches are available from the following URL:

`http://sunsolve.sun.com`

Caution – If you have not read the *Sun StorEdge QFS and Sun StorEdge SAM-FS 4.4 Release Notes*, please do so before continuing. You can access the *Sun StorEdge QFS and Sun StorEdge SAM-FS 4.4 Release Notes* for this release at any time from one of the documentation web sites described in this manual's preface.

▼ To Obtain the Software From the Sun Download Center

1. **Enter the following URL in your browser:**

   `http://www.sun.com/software/download/sys_admin.html`

2. **Click the Sun StorEdge QFS software package you want to download.**
3. **Follow the instructions on the web site for downloading the software.**

Software Licensing

You must agree to all binary and right-to-use (RTU) software license agreements before installing Sun StorEdge QFS software. Beginning with version 4, update 3, of the software, all media kits and software license options are delivered online, and license keys are no longer required.

Setting Up the Network Management Station

Read this section if you want to monitor your configuration through Simple Network Management Protocol (SNMP) software.

You can configure the Sun StorEdge QFS software to notify you when potential problems occur in its environment. The SNMP software manages information exchange between network devices such as servers, automated libraries, and drives. When the Sun StorEdge QFS software detects potential problems in its environment, it sends information to a management station, which enables you to monitor the system remotely.

The management stations you can use include the following:

- Sun Storage Automated Diagnostic Environment (StorADE)
- Sun Management Center (Sun MC)
- Sun Remote Server (SRS)
- Sun Remote Services Net Connect

If you want to enable SNMP traps, make sure that the management station software is installed and operating correctly before installing the Sun StorEdge QFS software. Refer to the documentation that came with your management station software for information on installation and use.

The types of problems, or events, that the Sun StorEdge QFS software can detect are defined in the Sun StorEdge QFS Management Information Base (MIB). The events include errors in configuration, `tapealert`(1M) events, and other atypical system activity. For complete information on the MIB, see `/opt/SUNWsamfs/mibs/SUN-SAM-MIB.mib` after the packages are installed.

The Sun StorEdge QFS software supports the TRAP SNMP (V2c) protocol. The software does not support `GET-REQUEST`, `GETNEXT-REQUEST`, and `SET_REQUEST`.

CHAPTER 3

Installation and Configuration Tasks

This chapter describes the procedures for installing and configuring Sun StorEdge QFS software for the first time. Use these procedures if this is the initial installation of the Sun StorEdge QFS software package at your site. If you are upgrading Sun StorEdge QFS software on an existing server, see Chapter 5, "Upgrade and Configuration Tasks" on page 75.

You can install and configure the Sun StorEdge QFS file system entirely using command-line interface (CLI) commands, or you can use a combination of CLI commands and the File System Manager browser interface tool.

If you are installing a Sun StorEdge QFS shared file system or a Sun StorEdge QFS file system in a Sun Cluster environment, you must also follow the additional installation instructions in the next chapter, "Configuration Tasks for a Shared or Sun Cluster Configuration" on page 57.

You must be logged in as superuser to complete most of the procedures in this chapter.

This chapter contains the following sections:

- "Installing the Software Packages" on page 24
- "Setting Up the Environment Configuration" on page 30
- "Setting Up Mount Parameters" on page 34
- "Initializing the Environment" on page 40
- "Performing Additional Configuration Tasks" on page 45
- "Backing Up Data" on page 54

Installing the Software Packages

The Sun StorEdge QFS software uses the Sun Solaris packaging utilities for adding and deleting software. The pkgadd(1M) utility prompts you to confirm various actions necessary to install the packages.

▼ To Add the Packages

Follow these steps on each host in the file system.

1. **Become superuser.**
2. **Use the** cd**(1) command to change to the directory where the software package release files reside.**

   If you are using a CD, the packages reside in the /cdrom/cdrom0 directory organized by Sun Solaris version.
3. **Use the** pkgadd**(1M) command to add the** SUNWqfsr **and** SUNWqfsu **packages.**

   For example:

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

4. **Enter** yes **or** y **as the answer to each of the questions.**

   When you install SUNWqfsr and SUNWqfsu, you are asked whether you want to define an administrator group. Select y to accept the default (no administrator group), or select n if you want to define an administrator group. You can reset permissions on certain commands later by using the set_admin(1M) command. For more information on this command, see "Adding the Administrator Group" on page 52 or the set_admin(1M) man page.
5. **Issue the** pkginfo**(1M) command and examine its output to make sure that a Sun StorEdge QFS package is installed.**

   Each host must have the SUNWqfsr and SUNWqfsu packages installed on it.

   CODE EXAMPLE 3-1 shows the needed SUNWqfsr/SUNWqfsu packages.

CODE EXAMPLE 3-1 pkginfo(1M) Command Example on a Sun StorEdge QFS File System

```
# pkginfo | grep SUNWqfs
system  SUNWqfsr     Sun QFS software Solaris 9 (root)
system  SUNWqfsu     Sun QFS software Solaris 9 (usr)
```

Installing the Linux Client Software

If you are installing the Sun StorEdge QFS software onto Linux clients in a shared environment, see the README file on Disk 1 of the Sun StorEdge QFS Linux Client installation package for instructions.

▼ To Set Up PATH and MANPATH Variables

In order to have access to the commands and man pages for the Sun StorEdge QFS commands, you must modify your PATH and MANPATH environment variables.

Follow these steps on each host in the fie system.

1. **For users who will need to access the Sun StorEdge QFS user commands (for example, sls(1)), add /opt/SUNWsamfs/bin to the users' PATH variables.**
2. **Use vi(1) or another editor to edit your system setup files to include the correct paths to commands and man pages.**
   - In the Bourne or Korn shell, edit the .profile file, change the PATH and MANPATH variables, and export the variables.

     CODE EXAMPLE 3-2 shows how your .profile file might look after editing.

CODE EXAMPLE 3-2 Finished .profile File

```
PATH=$PATH:/opt/SUNWsamfs/bin:/opt/SUNWsamfs/sbin
MANPATH=$MANPATH:/opt/SUNWsamfs/man
export PATH MANPATH
```

   - In the C shell, edit the .login and .cshrc files.

     When you have finished editing, the path statement in your .cshrc file might look like the following line:

```
set path = ($path /opt/SUNWsamfs/bin /opt/SUNWsamfs/sbin)
```

     CODE EXAMPLE 3-3 shows how the MANPATH in your .login file might look after you have finished editing.

CODE EXAMPLE 3-3 Finished MANPATH in the .login File

```
setenv MANPATH /usr/local/man:opt/SUNWspro/man:/$OPENWINHOME/\
share/man:/opt/SUNWsamfs/man
```

Installing and Using the File System Manager Software

Perform the tasks in this section if you want to be able to use the File System Manager software to configure, control, monitor, or reconfigure your Sun StorEdge QFS environment.

The procedures in this section are as follows:

- "To Install the File System Manager Software" on page 26.
- "Using the File System Manager Software" on page 28.

Note – File System Manager does not support file systems in Sun Cluster environments.

▼ To Install the File System Manager Software

1. **Ensure that you have met the installation requirements in "Verifying Requirements for File System Manager" on page 15.**
2. **Log in to the server that you want to use as the management station.**

   This can be the same server on which you installed the `SUNWqfsr` and `SUNWqfsu` packages.
3. **Become superuser.**
4. **Use the `cd`(1) command to change to the directory where the software package release files reside on your server.**
5. **Execute the `fsmgr_setup` script to start the installation process.**

   For example:

   ```
   # ./fsmgr_setup
   ```

6. **Answer the questions as prompted by the `fsmgr_setup` script.**

   During the installation procedure, you are asked questions about your environment. The script prompts you to enter passwords for the `SAMadmin` role and for the `samadmin` and `samuser` login IDs.

Note – When installing File System Manager, you must enter a password for all users and roles as prompted by the install script. If you leave a password blank, the Java Web Console will display an "Authentication Error" message when you try to log in.

The `fsmgr_setup` script automatically installs the following:

- The Tomcat, Java Runtime Environment (JRE), JATO, and Java Web Console packages. If you have existing versions of these software packages that are not compatible with File System Manager, the installation software asks you whether you want the appropriate levels to be installed at this time.
- The `SUNWfsmgru` package.
- The `SUNWfsmgrr` package.

The installation scripts prompt you to specify whether you want to install localized packages.

After installing the packages, it starts the Tomcat Web Server, enables logging, and creates the `SAMadmin` role.

7. **Use `vi`(1) or another editor to edit your system setup files to include the correct paths to commands and man pages.**

   - In the Bourne or Korn shell, edit the `.profile` file, change the `PATH` and `MANPATH` variables, and export the variables.

     CODE EXAMPLE 3-4 shows how your `.profile` file might look after editing.

CODE EXAMPLE 3-4 Finished `.profile` File

```
PATH=$PATH:/opt/SUNWfsmgr/bin
MANPATH=$MANPATH:/opt/SUNWfsmgr/man
export PATH MANPATH
```

   - In the C shell, edit the `.login` and `.cshrc` files.

     When you have finished editing, the `path` statement in your `.cshrc` file might look like the following line:

```
set path = ($path /opt/SUNWfsmgr/bin)
```

     CODE EXAMPLE 3-5 shows how the `MANPATH` in your `.login` file might look after you have finished editing.

CODE EXAMPLE 3-5 Finished `MANPATH` in the `.login` File

```
setenv MANPATH /usr/local/man:opt/SUNWspro/man:/$OPENWINHOME/\
share/man:/opt/SUNWsamfs/man:/opt/SUNWfsmgr/man
```

8. **Log in to the Sun StorEdge QFS server and become superuser.**

9. **Use the `ps`(1) and `grep`(1) commands to make sure that the `rpcbind` service is running.**

```
# ps -ef | grep rpcbind
```

10. **Examine the output from the preceding commands.**

    The output should contain a line similar to the following:

```
root   269     1  0   Feb 08 ?        0:06 /usr/sbin/rpcbind
```

    If `rpcbind` does not appear in the output, enter the following command to start the `rpcbind` service:

```
# /usr/sbin/rpcbind
```

11. **(Optional) Start the File System Manager (`fsmgmtd`) daemon.**

    If you did not choose to start the File System Manager daemon automatically during the installation process, do one of the following:

    - Enter the following command to start the File System Manager daemon and have it restart automatically every time the daemon process dies. With this configuration, the daemon also automatically restarts at system reboot.

```
# /opt/SUNWsamfs/sbin/fsmadm config -a
```

    - Enter the following command if you want the File System Manager daemon to run only once and not automatically restart.

```
# /opt/SUNWsamfs/sbin/fsmadm start
```

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

Using the File System Manager Software

After File System Manager is installed, you can log in to the software using two possible user names (`samadmin` and `samuser`) and two different roles (`SAMadmin` or `no role`). The tasks you can perform using the File System Manager differ as follows, depending on the user name and the role you assume at login:

- If you log in as `samadmin`, you can choose from one of the following roles:
  - The role of `SAMadmin` grants you full administrator privileges to configure, monitor, control, and reconfigure the devices in your Sun StorEdge QFS environment.

    Only the Sun StorEdge QFS administrator should log in using the `SAMadmin` role. All other users should log in as `samuser`.
  - The role of `no role` enables you only to monitor the environment. You cannot change or reconfigure it in any way.
- If you log in as `samuser`, you can only monitor the environment. You cannot change or reconfigure it in any way.

With regard to system administration, be aware that the Solaris OS root user on the server that hosts File System Manager is not necessarily the administrator of the File System Manager. Only `samadmin` has administrator privileges for the File System Manager application. The root user is the administrator of the management station.

By default, File System Manager is set up to manage the server on which it is installed. It can also be used to manage other servers running Sun StorEdge QFS software, but those additional servers must first be configured to allow File System Manager access. For instructions on adding additional managed servers, see the *Sun StorEdge QFS Configuration and Administration Guide* or the File System Manager online help.

▼ To Invoke File System Manager for the First Time

Perform this procedure if you want to invoke File System Manager and use it, rather than CLI commands, to perform some of the configuration steps.

1. **Log in to server where File System Manager is installed, or in to any computer that has network access to it.**
2. **If you upgraded from a previous version of the software, open the web browser and clear the browser cache.**
3. **From the web browser, invoke the File System Manager software.**

   The URL is as follows:

   ```
   https://hostname:6789
   ```

   For *hostname*, type the name of the host where the File System Manager software is installed. If you need to specify a domain name in addition to the host name, specify the *hostname* in this format: *hostname*.*domainname*.

   Note that this URL begins with `https`, not `http`. The Sun Java Web Console login screen appears.

4. **At the User Name prompt, enter** `samadmin`**.**
5. **At the Password prompt, enter the password you entered when you answered questions during the** `fsmgr_setup` **script's processing in "To Install the File System Manager Software" on page 26.**
6. **Click the** `SAMadmin` **role.**

Note – Only the Sun StorEdge QFS administrator should log in with the `SAMadmin` role.

---

7. **At the Role Password prompt, enter the password you entered in Step 5.**
8. **Click Log In.**
9. **In the Storage section, click File System Manager.**

   You are now logged in to the File System Manager interface.

   - If you want to configure your environment at this time using the File System Manager, stay on this page and add the server that you want to administer.

     If you need help accomplishing this task, click Help. After you add the servers, see the following sections for more information about configuring your environment using the File System Manager.
   - If you want to quit using the File System Manager at this time, click Log Out.

Setting Up the Environment Configuration

Each Sun StorEdge QFS environment is unique. The system requirements and hardware that are used differ from site to site. It is up to you, the system administrator at your site, to set up the specific configuration for your Sun StorEdge QFS environment.

The master configuration file, `/etc/opt/SUNWsamfs/mcf`, defines the topology of the equipment managed by the Sun StorEdge QFS file system. This file specifies the devices and file systems included in the environment and contains information that enables you to identify the disk slices to be used and to organize them into Sun StorEdge QFS file systems.

You can edit the `mcf` file in either of two ways:

- By using the File System Manager interface to configure Sun StorEdge QFS devices. When you create a file system using the File System Manager software, it creates an `mcf` file in `/etc/opt/SUNWsamfs/mcf` that contains a line for each device and family set of the file system.
- By directly editing the file using a text editor.

There are examples of `mcf` files in `/opt/SUNWsamfs/examples`. Example `mcf` file configurations can also be found in Appendix D of this manual.

Note – For information about file system design considerations, see the *Sun StorEdge QFS Configuration and Administration Guide*.

The following sections provide examples and describe activities related to creating and maintaining the `mcf` file:

- "To Create an `mcf` File Using the File System Manager Software" on page 31
- "To Create an `mcf` File Using a Text Editor" on page 32
- "To Verify the `mcf` File" on page 33

Note – The instructions in this section are for creating an `mcf` file in a Sun StorEdge QFS environment. If you are creating a SAM-QFS environment, follow the instructions in this section for configuring the file system portion of the `mcf` file. Then follow the instructions in the *Sun StorEdge SAM-FS Installation and Upgrade Guide* for library and drive configuration.

▼ To Create an `mcf` File Using the File System Manager Software

When you configure Sun StorEdge QFS file systems using the File System Manager software, it creates or edits the appropriate Sun StorEdge QFS configuration files, including the `mcf` file, on that server. You can use either File System Manager or the CLI to further edit these files later.

To add a server and create a file system:

1. **Log in to the File System Manager browser interface as an administrator user.**

   The Servers page is displayed.

2. **If you have not already added the server to be managed, click Add to add it now.**

   The Add Server page is displayed.

3. **In the Server Name or IP Address field, type the name of the server or the IP address of the server.**

4. **Click OK.**

5. **Click New File System.**

   The New File System wizard is displayed.

   Complete the steps for creating a new file system. When you complete this process, the `mcf` file is created. For more information, see the File System Manager online help.

▼ To Create an `mcf` File Using a Text Editor

- **Use `vi`(1) or another editor to create the `mcf` file.**

  When you create the `mcf` file, follow these guidelines:

  - Delimit the fields in each line with spaces or tabs.
  - Begin each comment line entered into this file with a pound sign (`#`).
  - Use a dash (–) to indicate optional fields that are omitted.
  - If you are creating a Sun StorEdge QFS shared file system, create the `mcf` file on the metadata server first.

  CODE EXAMPLE 3-6 shows the fields of each line entry in the `mcf` file.

CODE EXAMPLE 3-6 `mcf` File Fields

```
#
# Sun QFS file system configuration
#
# Equipment       Equip  Equip Fam   Dev    Additional
# Identifier      Ord    Type  Set   State  Parameters
# ----------      -----  ----- ----  -----  ----------
```

For more information, see the `mcf`(4) man page.

Caution – Be sure to specify disk partitions that are not in use on your system. Do not use overlapping partitions.

If you give the wrong partition names when creating any type of file system, you risk damaging user or system data. The risk is greatest if the partition named contains a UFS file system that is not mounted currently.

CODE EXAMPLE 3-7 shows file system entries in an `mcf` file for a Sun StorEdge QFS file system that is local to one Solaris OS host.

CODE EXAMPLE 3-7 Example Sun StorEdge QFS `mcf` File

```
#
# Sun QFS file system configuration
#
# Equipment       Equip  Equip Fam   Dev    Additional
# Identifier      Ord    Type  Set   State  Parameters
# ----------      -----  ----- ----  -----  ----------
qfs1               1     ma    qfs1  on
/dev/dsk/c1t0d0s0 11     mm    qfs1  on
/dev/dsk/c1t1d0s4 12     mr    qfs1  on
/dev/dsk/c1t2d0s4 13     mr    qfs1  on
/dev/dsk/c1t3d0s4 14     mr    qfs1  on
```

Note – If you change the `mcf` file after the Sun StorEdge QFS file system is in use, you must convey the new `mcf` specifications to the Sun StorEdge QFS software. For information about propagating `mcf` file changes to the system, see the *Sun StorEdge QFS Configuration and Administration Guide*.

For information about editing the `mcf` file on multiple hosts in a Sun StorEdge QFS shared file system or a file system in a Sun Cluster environment, see "Editing `mcf` Files on Other Hosts" on page 59.

▼ To Verify the `mcf` File

The procedures in this section show you how to verify the correctness of the `mcf` configuration file.

Perform these verifications on all hosts if you are configuring a Sun StorEdge QFS shared file system or a Sun StorEdge QFS highly available file system.

1. **Enter the `sam-fsd`(1M) command.**
2. **Examine the output for errors, as follows:**

- If the `mcf` file is free of syntax errors, the `sam-fsd`(1M) output is similar to that shown in CODE EXAMPLE 3-8. It contains information about the file systems and other system information.

CODE EXAMPLE 3-8 `sam-fsd`(1M) Output Showing No Errors

```
# sam-fsd
Trace file controls:
sam-amld      off

sam-archiverd off

sam-catserverd off

sam-fsd       off

sam-rftd      off

sam-recycler  off

sam-sharefsd  off

sam-stagerd   off

sam-serverd   off

sam-clientd   off

sam-mgmt      off
```

- If the `mcf` file contains syntax or other errors, the errors are noted in the output.

  If your `mcf` file has errors, refer to the `mcf`(4) man page for information about how to create this file correctly. You can also refer to the example `mcf` files in Appendix D.

Setting Up Mount Parameters

Use the procedures in this section to specify mount parameters for the Sun StorEdge QFS file system.

You can specify mount parameters in the following ways:

- On the `mount`(1M) command. Mount options specified here override those specified in the `/etc/vfstab` file and in the `samfs.cmd` file.

- In the /etc/vfstab file. Mount options specified here override those specified in the samfs.cmd file.
- In the samfs.cmd file.

Updating the /etc/vfstab File and Creating the Mount Point

This section describes how to edit the /etc/vfstab file.

Note – Even though /global is used in this chapter's examples as the mount point for file systems mounted in a Sun Cluster environment, it is not required. You can use any mount point.

TABLE 3-1 shows the values you can enter in the fields in the /etc/vfstab file.

TABLE 3-1 Fields in the /etc/vfstab File

Field	Field Title and Contents
1	Device to Mount. The name of the Sun StorEdge QFS file system to be mounted. This must be the same as the file system's Family Set name specified in the mcf file.
2	Device to fsck(1M). Must be a dash (-) character, which indicates that there are no options. This prevents the Solaris system from performing an fsck(1M) process on the Sun StorEdge QFS file system. For more information about this process, see the fsck(1M) or samfsck(1M) man page.
3	Mount Point. For example: • /qfs1 for a local Sun StorEdge QFS file system on a single host. • /global/qfs1 for a Sun StorEdge QFS shared file system in a Sun Cluster environment. • /global/qfs1 for a Sun StorEdge QFS highly available file system in a Sun Cluster environment.
4	File System Type. Must be samfs.

TABLE 3-1 Fields in the `/etc/vfstab` File *(Continued)*

Field	Field Title and Contents
5	`fsck`(1M) Pass. Must be a dash (-) character, which indicates that there are no options.
6	Mount at Boot. Either `yes` or `no`. • Specifying `yes` in this field indicates that the Sun StorEdge QFS file system is to be mounted automatically at boot time. Do not specify `yes` if you are creating a file system for use in a Sun Cluster environment. • Specifying `no` in this field indicates that you do not want to mount the file system automatically. Specify `no` in this field if you are creating a file system for use in a Sun Cluster environment to indicate that the file system is under Sun Cluster software control. For information about the format of these entries, see the `mount_samfs`(1M) man page.
7	Mount Parameters. A list of comma-separated parameters (with no spaces) that are used in mounting the file system. You can specify mount options on the `mount`(1M) command, in the `/etc/vfstab` file, or in a `samfs.cmd` file. Mount options specified on the `mount`(1M) command override those specified in the `/etc/vfstab` file and in the `samfs.cmd` file. Mount options specified in the `/etc/vfstab` file override those in the `samfs.cmd` file. For example, `stripe=1` specifies a stripe width of one DAU. For a list of available mount options, see the `mount_samfs`(1M) man page.

▼ To Update the `/etc/vfstab` File Using File System Manager

When you create a file system using File System Manager, a default `/etc/vfstab` file is created. However, mount options specified in File System Manager are written to the `samfs.cmd` file rather than to the `/etc/vfstab` file. For more information, see "To Create and Edit the `samfs.cmd` File Using File System Manager" on page 39.

To edit the mount options in the `/etc/vfstab` file, use the following command-line procedure, "To Update the `/etc/vfstab` File and Create the Mount Point Using a Text Editor" on page 36.

▼ To Update the `/etc/vfstab` File and Create the Mount Point Using a Text Editor

Follow these steps on all hosts if you are configuring a Sun StorEdge QFS shared file system or a Sun StorEdge QFS highly available file system.

For debugging purposes, if you are configuring a Sun StorEdge QFS shared file system, the mount options should be the same on all hosts that can mount the file system.

1. **Use `vi`(1) or another editor to open the `/etc/vfstab` file.**

2. **Create an entry for each Sun StorEdge QFS file system.**

   CODE EXAMPLE 3-9 shows header fields and entries for a local Sun StorEdge QFS file system.

CODE EXAMPLE 3-9 Example `/etc/vfstab` File Entries for a Sun StorEdge QFS File System

```
#DEVICE    DEVICE   MOUNT   FS     FSCK   MOUNT     MOUNT
#TO MOUNT  TO FSCK  POINT   TYPE   PASS   AT BOOT   PARAMETERS
#
qfs1       -        /qfs1   samfs  -     yes       stripe=1
```

TABLE 3-1 shows the various fields in the `/etc/vfstab` file and their contents.

If you are configuring a file system for a Sun Cluster environment, the required and recommended mount options vary according to the type of file system you are configuring. TABLE 3-2 explains the mount options.

TABLE 3-2 Mount Options for a Sun Cluster File System

File System Type	Required Options	Recommended Options

Sun StorEdge QFS shared file system	`shared`	`forcedirectio`
		`sync_meta=1`
		`mh_write`
		`qwrite`
		`nstreams=1024`
		`rdlease=300`
		`aplease=300`
		`wrlease=300`

Sun StorEdge QFS shared file system to support Oracle Real Application Clusters database files	`shared`
	`forcedirectio`
	`sync_meta=1`
	`mh_write`
	`qwrite`
	`nstreams=1024`
	`stripe>=1`
	`rdlease=300`
	`aplease=300`
	`wrlease=300`	

Sun StorEdge QFS highly available file system	None	`sync_meta=1`

You can specify most of the mount options listed in TABLE 3-2 in either the `/etc/vfstab` file or the `samds.cmd` file. The one exception is the `shared` option, which must be specified in the `/etc/vfstab` file.

Tip – In addition to the mount options mentioned in TABLE 3-2, you can also specify the `trace` mount option for configuration debugging purposes.

---

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

   The mount point location varies, depending on where the file system is to be mounted. The following examples illustrate this.

   Example 1. This example assumes that `/qfs1` is the mount point of the `qfs1` file system. This is a local file system. It can exist on a standalone server or on a local node in a Sun Cluster environment.

```
# mkdir /qfs1
```

   Example 2. This example assumes that `/global/qfs1` is the mount point of the `qfs1` file system, which is a Sun StorEdge QFS shared file system to be mounted in a Sun Cluster environment.

```
# mkdir /global/qfs1
```

Note – If you configured multiple mount points, repeat these steps for each mount point, using a different mount point (such as `/qfs2`) and Family Set name (such as `qfs2`) each time.

Creating and Editing the `samfs.cmd` File

You can create the `/etc/opt/SUNWsamfs/samfs.cmd` file as the place from which the system reads mount parameters.

You can manage certain features more easily from a `samfs.cmd` file. These features include the following:

- Striping.
- Readahead, which specifies the number of bytes that are read ahead during performance of paged I/O.
- Writebehind, which specifies the number of bytes that are written behind during performance of paged I/O.

- Qwrite, which enables simultaneous reads and writes to the same file from different threads.

If you are configuring multiple Sun StorEdge QFS systems with multiple mount parameters, consider creating the `samfs.cmd` file.

▼ To Create and Edit the `samfs.cmd` File Using File System Manager

If you specify non-default mount options when creating a file system in File System Manager, the `samfs.cmd` file is automatically created or updated with those mount options.

To edit a file system's mount options:

1. **From the Servers page, click the name of the server on which the file system is located.**

   The File Systems Summary page is displayed.

2. **Select the radio button next to the file system whose mount options you want to edit.**

3. **From the Operations menu, choose Edit Mount Options.**

   The Edit Mount Options page is displayed.

4. **Make your edits in the fields.**

   For more information about the fields on the Edit Mount Options page, see the File System Manager online help.

5. **Click Save.**

▼ To Create and Edit the `samfs.cmd` File Using a Text Editor

1. **Use `vi`(1) or another editor to create the `samfs.cmd` file.**

   Create lines in the `samfs.cmd` file to control mounting, performance features, or other aspects of file system management. For more information about the `samfs.cmd` file, see the *Sun StorEdge QFS Configuration and Administration Guide*, or see the `samfs.cmd`(4) man page.

   shows a `samfs.cmd` file for a Sun StorEdge QFS file system.

CODE EXAMPLE 3-10 Example `samfs.cmd` File

```
low = 50
high = 75
fs = samfs1
   high = 65
   writebehind = 512
```

CODE EXAMPLE 3-10 Example `samfs.cmd` File *(Continued)*

```
    readahead = 1024
fs = samfs5
    partial = 64
```

2. **If you are creating a multihost file system, copy lines as necessary to the `samfs.cmd` file on other hosts.**

   If you have created a `samfs.cmd` file on one host in a Sun Cluster environment to describe a particular file system's mount parameters, copy those lines to the `samfs.cmd` files on all the nodes that can access that file system.

   For debugging purposes, the `samfs.cmd` file, as it pertains to a specific file system, should be the same on all hosts. For example, if the `qfs3` file system is accessible from all nodes in a Sun Cluster environment, then the lines in the `samfs.cmd` file that describe the `qfs3` file system should be identical on all the nodes in the Sun Cluster environment.

   Depending on your site needs, it might be easier to manage mount options from the `samfs.cmd` file rather than from the `/etc/vfstab` file. The `/etc/vfstab` file overrides the `samfs.cmd` file in the event of conflicts.

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

Initializing the Environment

This section tells you how to initialize the environment and the Sun StorEdge QFS file system and how to mount the file system.

▼ To Initialize the Environment

- **Use the `samd`(1M) `config` command to initialize the Sun StorEdge QFS environment:**

```
# samd config
```

Repeat this command on each host if you are configuring a Sun StorEdge QFS shared file system or a Sun StorEdge QFS highly available file system.

▼ To Initialize the File System

This procedure shows how to use the `sammkfs`(1M) command and the Family Set names that you have defined to initialize a file system.

Note – The `sammkfs`(1M) command sets one tuning parameter, the disk allocation unit (DAU). You cannot reset this parameter without reinitializing the file system. For information about how the DAU affects tuning, see the *Sun StorEdge QFS Configuration and Administration Guide* or see the `sammkfs`(1M) man page.

- **Use the `sammkfs`(1M) command to initialize a file system for each Family Set name defined in the `mcf` file.**

Caution – Running `sammkfs`(1M) creates a new file system. It removes all references to the data currently contained in the partitions associated with the file system in the `/etc/opt/SUNWsamfs/mcf` file.

Initializing a Sun StorEdge QFS File System

CODE EXAMPLE 3-11 shows the command to use to initialize a Sun StorEdge QFS file system with the Family Set name of `qfs1`.

CODE EXAMPLE 3-11 Initializing Example File System `qfs1`

```
# sammkfs -a 128 qfs1
Building 'qfs1' will destroy the contents of devices:
                /dev/dsk/c1t0d0s0
                /dev/dsk/c3t1d0s6
                /dev/dsk/c3t1d1s6
                /dev/dsk/c3t2d0s6
Do you wish to continue? [y/N]
```

Enter `y` in response to this message to continue the file system creation process.

Initializing a Sun StorEdge QFS Shared File System

If you are configuring a Sun StorEdge QFS shared file system, enter the `sammkfs`(1M) command on the metadata server only.

Enter the sammkfs(1M) command at the system prompt. The -S option specifies that the file system be a Sun StorEdge QFS shared file system. Use this command in the following format:

```
sammkfs -S -a allocation-unit fs-name
```

TABLE 3-3 sammkfs(1M) Command Arguments

Argument	Meaning
allocation-unit	The number of bytes, in units of 1024 (1-kilobyte) blocks, to be allocated to a disk allocation unit (DAU). The specified *allocation-unit* value must be a multiple of 8 kilobytes. For more information, see the *Sun StorEdge QFS Configuration and Administration Guide* or the sammkfs(1M) man page.
fs-name	The Family Set name of the file system as defined in the mcf file.

For example, you might use the following sammkfs(1M) command to initialize a Sun StorEdge QFS shared file system and identify it as shared:

```
# sammkfs -S -a 512 sharefs1
```

If the shared keyword appears in the mcf file, the file system must be initialized as a shared file system with the -S option to the sammkfs(1M) command. You cannot mount a file system as shared if it was not initialized as shared.

If you are initializing a file system as a Sun StorEdge QFS shared file system, file /etc/opt/SUNWsamfs/hosts.*fs-name* must already exist when you issue the sammkfs(1M) command. The sammkfs(1M) command uses the hosts file when it creates the file system. You can use the samsharefs(1M) command to replace or update the contents of the hosts file at a later date. For more information, see "Creating the Shared Hosts File" on page 63.

Mounting the File System

The mount(1M) command mounts a file system. It also reads the /etc/vfstab and samfs.cmd configuration files. For information about the mount(1M) command, see the mount_samfs(1M) man page.

Use one or more of the procedures in this section to mount your file system.

▼ To Mount the File System Using File System Manager

1. **From the Servers page, click the name of the server on which the file system is located.**

   The File Systems Summary page is displayed.

2. **Select the radio button next to the file system that you want to mount.**

3. **From the Operations menu, choose Mount.**

▼ To Mount the File System on One Host From the Command Line

Perform this procedure on all Sun StorEdge QFS file systems, as follows:

- If you are configuring a Sun StorEdge QFS file system on a single host, perform this procedure on that host.
- If you are configuring a Sun StorEdge QFS shared file system in a Solaris OS environment, perform this procedure first on the metadata server and then on the other hosts in the file system.
- If you are configuring a Sun StorEdge QFS shared or highly available file system in a Sun Cluster environment, perform this procedure on all nodes that can host the file system.

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

   Specify the file system mount point as the argument. For example:

```
# mount /qfs1
```

2. **Use the `mount`(1M) command with no arguments to verify the mount.**

   This step confirms that the file system is mounted and shows how to set permissions. CODE EXAMPLE 3-12 shows the output from a `mount`(1M) command issued to verify whether example file system `qfs1` is mounted.

CODE EXAMPLE 3-12 Using the `mount`(1M) Command to Verify That a File System Is Mounted

```
# mount
<<< information deleted >>>
/qfs1 on qfs1 read/write/setuid/dev=8001b1 on Mon Jan 14 12:21:03 2002
<<< information deleted >>>
```

3. **(Optional) Use the `chmod`(1) and `chown`(1) commands to change the permissions and ownership of the file system's root directory.**

   If this is the first time the file system has been mounted, it is typical to perform this step.

For example:

```
# chmod 755 /qfs1
# chown root:other /qfs1
```

▼ To Verify Metadata Server Changes

If you are creating a Sun StorEdge QFS shared file system in either a Solaris OS or a Sun Cluster environment, perform this procedure to verify that the file system is configured so that you can change the metadata server.

If you are creating a Sun StorEdge QFS shared file system in a Solaris OS environment, perform these steps on each metadata server or potential metadata server.

If you are creating a Sun StorEdge QFS shared file system in a Sun Cluster environment, perform these steps on all hosts that can mount the file system.

1. **Log in to the metadata server as superuser.**
2. **Use the `samsharefs`(1M) command to change the metadata server.**

   For example:

   ```
   ash# samsharefs -s oak qfs1
   ```

3. **Use the `ls`(1) `-al` command to verify that the files are accessible on the new metadata server.**

   For example:

   ```
   oak# ls -al /qfs1
   ```

4. **Repeat Step 2 and Step 3 to change back to the original metadata server.**

For more information on completing the configuration of file systems in a shared or Sun Cluster Environment, see "Configuration Tasks for a Shared or Sun Cluster Configuration" on page 57 after completing the rest of the rest of the procedures in this chapter as needed.

Performing Additional Configuration Tasks

This section outlines additional tasks that you might need to complete in order to finish the configuration of the Sun StorEdge QFS environment. Some of these tasks are optional, depending on your specific environment. The following tasks are covered:

- "Sharing the File System With NFS Client Systems" on page 45
- "To Change Default Values" on page 48
- "Configuring the Remote Notification Facility" on page 49
- "Adding the Administrator Group" on page 52
- "To Enable Logging" on page 52
- "Configuring Other Products" on page 53

Sharing the File System With NFS Client Systems

Perform this task if you are configuring a file system and you want the file system to be NFS shared.

The procedures in this section use the Sun Solaris `share`(1M) command to make the file system available for mounting by remote systems. The `share`(1M) commands are typically placed in the `/etc/dfs/dfstab` file and are executed automatically by the Sun Solaris OS when you enter `init`(1M) state 3.

▼ To NFS-Share the File System in a Sun Cluster Environment

The following is a general description of how to NFS-share a file system in a Sun Cluster environment. For more information on NFS-sharing file systems that are controlled by HA Storage Plus, see the *Sun StorEdge QFS Configuration and Administration Guide*, the *Sun Cluster Data Service for Network File System (NFS) Guide for Solaris OS*, and your NFS documentation.

1. **Locate the** `dfstab.`*resource-name* **file.**

   The `Pathprefix` property of HA StorageHA Storage Plus specifies the directory in which the `dfstab.`*resource-name* file resides.

2. **Use `vi`(1) or another editor to add a `share`(1M) command to the *Pathprefix*`/SUNW.nfs/dfstab.`*resource-name* file.**

   For example:

```
share -F nfs -o rw /global/qfs1
```

▼ To NFS-Share the File System in a Solaris OS Environment

If you are configuring a Sun StorEdge QFS shared file system, you can perform this procedure from the metadata server or from one of the shared clients.

1. **Use `vi`(1) or another editor to add a `share`(1M) command to the `/etc/dfs/dfstab` file.**

   For example, add a line like the following to direct the Solaris OS to NFS share the new Sun StorEdge QFS file system:

```
share -F nfs -o rw=client1:client2 -d "QFS" /qfs1
```

2. **Use the `ps`(1) and `grep`(1) commands to determine whether `nfs.server` is running.**

   For example:

```
# ps -ef | grep nfsd
    root   694     1  0   Apr 29 ?         0:36 /usr/lib/nfs/nfsd -a 16
en17     29996 29940  0 08:27:09 pts/5     0:00 grep nfsd
# ps -ef | grep mountd
    root   406     1  0   Apr 29 ?        95:48 /usr/lib/autofs/automountd
    root   691     1  0   Apr 29 ?         2:00 /usr/lib/nfs/mountd
en17     29998 29940  0 08:27:28 pts/5     0:00 grep mountd
```

   In this sample output, the lines that contain `/usr/lib/nfs` indicate that the NFS server is mounted.

3. **If `nfs.server` is not running, start it.**

   For example:

```
# /etc/init.d/nfs.server start
```

4. **(Optional) Type the `share`(1M) command at a `root` shell prompt.**

   Perform this step if you want to NFS-share the new Sun StorEdge QFS file system immediately.

When you have finished NFS-sharing the file system, you are ready to mount it, as described in the next procedure, "To Mount the File System on NFS Clients in a Solaris OS Environment" on page 47.

Notes on NFS-Sharing

If no NFS-shared file systems exist when the Sun Solaris OS boots, the NFS server is not started. CODE EXAMPLE 3-13 shows the commands to use to enable NFS-sharing. You must change to run level 3 after adding the first share entry to this file.

CODE EXAMPLE 3-13 NFS Commands

```
# init 3
# who -r
.       run-level 3  Dec 12 14:39     3    2  2
# share
-           /qfs1   -   "QFS"
```

Some NFS mount parameters can affect the performance of an NFS-mounted Sun StorEdge QFS file system. You can set these parameters in the `/etc/vfstab` file as follows:

- `timeo = n`. This value sets the NFS timeout to *n* tenths of a second. The default is 11. For optimal performance, use the default value. You can increase or decrease the value as appropriate for your system.
- `rsize = n`. This value sets the read buffer size to *n* bytes. In NFS 2, change the default value (8192) to 32768. In NFS 3, retain the default value of 32768.
- `wsize = n`. This value sets the write buffer size to *n* bytes. In NFS 2, change the default value (8192) to 32768. In NFS 3, retain the default value of 32768.

For more information about these parameters, see the `mount_nfs`(1M) man page.

▼ To Mount the File System on NFS Clients in a Solaris OS Environment

If you are configuring a Sun StorEdge QFS shared file system, you can perform this procedure from the metadata server or from one of the shared clients.

Note – In Sun StorEdge QFS shared file systems, there can sometimes be a significant delay in the file system's response to NFS client requests. As a consequence, the system might generate an error instead of retrying the operation.

To avoid this situation, it is recommended that you mount the file system on NFS clients with either the `hard` option enabled or with the `soft`, `retrans`, and `timeo` options enabled. If you use the `soft` option, also specify `retrans=120` (or greater) and `timeo=3000` (or greater). These mount options can be specified in the `/etc/vfstab` file, as shown in the procedure below.

1. **On each NFS client system, use `vi`(1) or another editor to edit the `/etc/vfstab` file, and add a line to mount the server's Sun StorEdge QFS file system at a convenient mount point.**

   The following example mounts `server:/qfs1` on the `/qfs1` mount point:

```
server:/qfs1    -    /qfs1    nfs    -    no intr,timeo=60
```

2. **Save and close the `/etc/vfstab` file.**

3. **Enter the `mount`(1M) command.**

   The following `mount`(1M) command mounts the `qfs1` file system:

```
client# mount /qfs1
```

   Alternatively, the automounter can do this, if you prefer. Follow your site procedures for adding `server:/qfs1` to your automounter maps. For more information about automounting, see the `automountd`(1M) man page.

▼ To Change Default Values

The `/opt/SUNWsamfs/examples/defaults.conf` file contains default settings for the Sun StorEdge QFS environment. You can change these settings at any time after the initial installation.

Before changing any default settings, examine the `defaults.conf`(4) man page to discern the types of behavior this file controls.

Follow these steps for each host that you want to include in a Sun StorEdge QFS shared file system or a Sun StorEdge QFS highly available file system.

Note – For debugging purposes, the `defaults.conf` file should be the same on all hosts.

1. **Use the `cp`(1) command to copy `/opt/SUNWsamfs/examples/defaults.conf` to its functional location.**

   For example:

```
# cp /opt/SUNWsamfs/examples/defaults.conf /etc/opt/SUNWsamfs/defaults.conf
```

2. **Use `vi`(1) or another editor to edit the file.**

   Edit the lines that control aspects of the system that you want to change. Remove the pound character (#) from column 1 of the lines you change.

   For example, if you are configuring a Sun StorEdge QFS shared file system in a Sun Cluster environment, you might specify the following settings, which are helpful during debugging:

```
# File defaults.conf
trace
all=on
endtrace
```

3. **Use the `samd`(1M) `config` command to restart the `sam-fsd`(1M) daemon and enable the daemon to recognize the changes in the `defaults.conf` file.**

Configuring the Remote Notification Facility

The Sun StorEdge QFS software can be configured to notify you when potential problems occur in its environment. The system sends notification messages to a management station of your choice. The Simple Network Management Protocol (SNMP) software manages the exchange of information between network devices such as servers, automated libraries, and drives.

The Sun StorEdge QFS Management Information Base (MIB) defines the types of problems, or events, that the Sun StorEdge QFS software can detect. The software can detect errors in configuration, `tapealert`(1M) events, and other atypical system activity. For complete information about the MIB, see `/opt/SUNWsamfs/mibs/SUN-SAM-MIB.mib`.

The following procedures describe how to enable and disable remote notification.

▼ To Enable Remote Notification

1. **Ensure that the management station is configured and known to be operating correctly.**

   "Setting Up the Network Management Station" on page 20 describes this prerequisite.

2. **Using `vi`(1) or another editor, examine the `/etc/hosts` file to ensure that the management station to which notifications should be sent is defined. If it is not defined, add a line that defines the appropriate host.**

   The following sample file defines a management station with a host name of `mgmtconsole`.

   CODE EXAMPLE 3-14 Example `/etc/hosts` File

   ```
   999.9.9.9        localhost
   999.999.9.999    loggerhost      loghost
   999.999.9.998    mgmtconsole
   999.999.9.9      samserver
   ```

3. **Save your changes to `/etc/hosts` and exit the file.**

4. **Using `vi`(1) or another editor, open the file `/etc/opt/SUNWsamfs/scripts/sendtrap` and locate the `TRAP_DESTINATION=‘hostname‘` directive.**

   This line specifies that remote notification messages be sent to port 161 of the server upon which the Sun StorEdge QFS software is installed. Note the following:

   - If you want to change the host name or/and port, replace the `TRAP_DESTINATION` directive line with `TRAP_DESTINATION="`*mgmt-console-name*`:`*port*`"`. Note the use of quotation marks (`" "`) rather than apostrophes (`‘ ‘`) in the new directive.
   - If you want to send remote notification messages to multiple hosts, specify the directive in the following format:

   ```
   TRAP_DESTINATION="mgmt-console-name:port [ mgmt_console_name:port]"
   ```

   For example:

   ```
   TRAP_DESTINATION="localhost:161 doodle:163 mgmt_station:1162"
   ```

5. **Locate the** `COMMUNITY="public"` **directive in** `/etc/opt/SUNWsamfs/scripts/sendtrap`**.**

   This line acts as a password. It prevents unauthorized viewing or use of SNMP trap messages. Examine this line and do one of the following, depending on the community string value of your management station:

   - If your management station's community string is also set to `public`, you do not have to edit this value.
   - If your management station's community string is set to a value other than `public`, edit the directive to replace `public` with the value that is used in your management station.

6. **Save your changes to** `/etc/opt/SUNWsamfs/scripts/sendtrap` **and exit the file.**

▼ To Disable Remote Notification

The remote notification facility is enabled by default. If you want to disable remote notification, perform this procedure.

1. **If the file** `/etc/opt/SUNWsamfs/defaults.conf` **does not already exist, use the** `cp`**(1) command to copy file** `/opt/SUNWsamfs/examples/defaults.conf` **to** `/etc/opt/SUNWsamfs/defaults.conf`**.**

2. **Using** `vi`**(1) or another editor, open the file** `/etc/opt/SUNWsamfs/defaults.conf` **and find the line that specifies SNMP alerts.**

   The line is as follows:

   ```
   #alerts=on
   ```

3. **Edit the line to disable SNMP alerts.**

   Remove the `#` symbol and change `on` to `off`. After editing, the line is as follows:

   ```
   alerts=off
   ```

4. **Save your changes and exit the file.**

5. **Use the** `samd`**(1M)** `config` **command to restart the** `sam-fsd`**(1M) daemon.**

   ```
   # samd config
   ```

   This command restarts the `sam-fsd`(1M) daemon and enables the daemon to recognize the changes in the `defaults.conf` file.

Adding the Administrator Group

By default, only the superuser can execute Sun StorEdge QFS administrator commands. However, during installation you can create an administrator group. Members of the administrator group can execute all administrator commands except for `star`(1M), `samfsck`(1M), `samgrowfs`(1M), `sammkfs`(1M), and `samd`(1M). The administrator commands are located in `/opt/SUNWsamfs/sbin`.

After installing the package, you can use the `set_admin`(1M) command to add or remove the administrator group. You must be logged in as superuser to use the `set_admin`(1M) command. You can also undo the effect of this selection and make the programs in `/opt/SUNWsamfs/sbin` executable only by the superuser. For more information about this command, see the `set_admin`(1M) man page.

▼ To Add the Administrator Group

1. **Choose an administrator group name, or select a group that already exists in your environment.**
2. **Use the `groupadd`(1M) command, or edit the `/etc/group` file.**

   The following is an entry from the `/etc/group` file that designates an administrator group for the Sun StorEdge QFS software. In this example, the `samadm` group consists of both the `adm` and `operator` users.

```
samadm::1999:adm,operator
```

▼ To Enable Logging

The Sun StorEdge QFS system logs errors, cautions, warnings, and other messages using the standard Sun Solaris `syslog`(3) interface. By default, the Sun StorEdge QFS facility is `local7`.

1. **Use `vi`(1) or another editor to open the `/etc/syslog.conf` file.**
2. **In the file `/opt/SUNWsamfs/examples/syslog.conf_changes`, find the logging line, which is similar, if not identical, to the following:**

```
local7.debug  /var/adm/sam-log
```

Note – The preceding entry is all one line and has a TAB character (not a space) between the fields.

The default facility is `local7`. If you set logging to something other than `local7` in the `/etc/syslog.conf` file, edit the `defaults.conf` file and reset it there, too. For more information, see the `defaults.conf`(4) man page.

3. **Append the logging line from `/opt/SUNWsamfs/examples/syslog.conf_changes` to the `/etc/syslog.conf` file.**

   For example:

```
# cp /etc/syslog.conf /etc/syslog.conf.orig
# cat /opt/SUNWsamfs/examples/syslog.conf_changes >> /etc/syslog.conf
```

4. **Create an empty log file and send the `syslogd` process a HUP signal.**

   For example, to create a log file in `/var/adm/sam-log` and send the HUP to the `syslogd` daemon, type the following:

```
# touch /var/adm/sam-log
# pkill -HUP syslogd
```

   For more information, see the `syslog.conf`(4) and `syslogd`(1M) man pages.

5. **(Optional) Use the `log_rotate.sh`(1M) command to enable log file rotation.**

   Log files can become very large, and the `log_rotate.sh`(1M) command can help in managing log files. For more information, see the `log_rotate.sh`(1M) man page.

Configuring Other Products

The Sun StorEdge QFS installation and configuration process is complete. You can configure other Sun products at this time.

For example, if you want to configure an Oracle database, see the *Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS*. The Oracle Real Application Clusters application is the only scalable application that the Sun StorEdge QFS software supports in Sun Cluster environments.

Backing Up Data

This section describes the recommended procedures for regularly backing up important data and files in a Sun StorEdge QFS environment.

Setting Up Dump Files

File systems are made up of directories, files, and links. The Sun StorEdge QFS file system keeps track of all the files in the `.inodes` file. The `.inodes` file resides on a separate metadata device. The file system writes all file data to the data devices.

It is important to use the `qfsdump`(1M) command periodically to create a dump file of metadata and file data. The `dump` process saves the relative path information for each file contained in a complete file system or in a portion of a file system. This protects your data in the event of a disaster.

You can create dump files as often as once or twice a day, depending on your site's requirements. By dumping file system data on a regular basis, you can restore old files and file systems. You can also move files and file systems from one server to another.

The following are some guidelines for creating dump files:

- The `qfsdump`(1M) command dumps file names, inode information, and data. This command creates full dumps, not incremental dumps, of specified files and directories, and the resulting file can therefore be very large. The `qfsdump`(1M) command does not have any tape management, size estimations, or incremental dump facilities as does `ufsdump`(1M). In addition, the `qfsdump`(1M) command does not support volume overflow, so you need to weigh space considerations and make sure that the size of the file system does not exceed the size of the dump media.
- The `qfsdump`(1M) command dumps all the data of a sparse file, and the `qfsrestore`(1M) command restores all the data. These commands do not, however, preserve file qualities that enable sparse files to be characterized as sparse. This can cause files to occupy more space on dump files and on restored file systems than anticipated.
- Because you issue the `qfsdump`(1M) command on a mounted file system, inconsistencies can arise as new files are being created on disk. Dumping file systems during a quiet period (a time when files are not being created or modified) is a good idea and minimizes these inconsistencies.
- Ensure that you dump metadata and data for all Sun StorEdge QFS file systems. Look in `/etc/vfstab` for all file systems of type `samfs`.

You can run the qfsdump(1M) command manually or automatically. Even if you implement this command to be run automatically, you might also need to run it manually from time to time, depending on your site's circumstances. In the event of a disaster, you can use the qfsrestore(1M) command to re-create your file system. You can also restore a single directory or file. For more information, see the qfsdump(1M) man page and see the *Sun StorEdge SAM-FS Troubleshooting Guide*.

For more information about creating dump files, see the qfsdump(1M) man page. The following sections describe procedures for issuing this command both manually and automatically.

▼ To Run the qfsdump(1M) Command Automatically Using cron

Perform this step for each Sun StorEdge QFS file system in your environment. Make sure you save each dump file in a separate file.

- **For each file system, make an entry in the root crontab file so that the cron daemon runs the qfsdump(1M) command periodically.**

  For example:

```
10 0 * * * (cd /qfs1; /opt/SUNWsamfs/sbin/qfsdump -f /dev/rmt/0cbn)
```

  This entry executes the qfsdump(1M) command at 10 minutes after midnight. It uses the cd(1) command to change to the mount point of the qfs1 file system, and it executes the /opt/SUNWsamfs/sbin/qfsdump command to write the data to tape device /dev/rmt/0cbn.

▼ To Run the qfsdump(1M) Command Manually From the Command Line

1. **Use the cd(1) command to go to the directory that contains the mount point for the file system.**

   For example:

```
# cd /qfs1
```

2. **Use the qfsdump(1M) command to write a dump file to a file system outside of the one you are dumping.**

   For example:

```
# qfsdump -f /save/qfs1/dump_file
```

Backing Up Configuration Files

The Sun StorEdge QFS software regularly accesses several files that have been created as part of this installation and configuration procedure. You should back up these files regularly to a file system that is outside the file system in which they reside. In the event of a disaster, you can then restore these files from your backup copies.

The following files are among those that you should back up regularly and whenever you modify them:

- `/etc/opt/SUNWsamfs/mcf`
- `/etc/opt/SUNWsamfs/samfs.cmd`
- `/etc/opt/SUNWsamfs/defaults.conf`
- `/etc/opt/SUNWsamfs/archiver.cmd`

For more information about the files you should protect, see the *Sun StorEdge SAM-FS Troubleshooting Guide*.

CHAPTER 4

Configuration Tasks for a Shared or Sun Cluster Configuration

This chapter contains instructions for configuring the Sun StorEdge QFS software in a shared or Sun Cluster environment. Before carrying out the configuration procedures in this chapter, you must have installed the software as described in Chapter 3.

This chapter contains the following sections:

- "Preparing the Host Systems" on page 57
- "Editing `mcf` Files on Other Hosts" on page 59
- "Creating the Shared Hosts File" on page 63
- "Verifying That the Daemons Are Running" on page 69
- "Configuring the `SUNW.qfs` Resource Type" on page 70
- "Configuring the HA Storage Plus Resource" on page 72
- "Bringing the Shared Resource Online" on page 72
- "Verifying the Resource Group on All Nodes" on page 73

Preparing the Host Systems

Perform this procedure to prepare the host systems for a Sun StorEdge QFS shared file system or for a Sun StorEdge QFS shared file system in a Sun Cluster environment.

▼ To Prepare the Host Systems

1. **Verify that all the hosts have the same user and group IDs.**

   If you are not running the Network Information Name service (NIS), make sure that all `/etc/passwd` and all `/etc/group` files are identical. If you are running NIS, the `/etc/passwd` and `/etc/group` files should already be identical.

   For more information about this, see the `nis+`(1) man page.

2. **If you are configuring a Sun StorEdge QFS shared file system on the Solaris OS, enable the network time daemon command,** `xntpd`**(1M), to synchronize the times on all the hosts.**

   You do not need to perform this step if you are configuring a Sun StorEdge QFS shared file system in a Sun Cluster environment because it has already been done as part of the Sun Cluster installation.

   The clocks of all hosts must be synchronized, and must be kept synchronized, during Sun StorEdge QFS shared file system operations. For more information, see the `xntpd`(1M) man page.

   The following steps enable the `xntpd`(1M) daemon on one host. Follow these steps for each host.

   a. **Stop the** `xntpd`**(1M) daemon.**

      For example:

```
# /etc/init.d/xntpd stop
```

   b. **Use** `vi`**(1) or another editor to create the file** `/etc/inet/ntp.conf`**.**

   c. **Create a line in the file** `/etc/inet/ntp.conf` **that specifies the name of the local time server.**

      This line has the following format:

```
server IP-address prefer
```

      In the preceding command, `server` and `prefer` are required keywords. Specify the IP address of your local time server for *IP-address*.

      If you have no local time server, see one of the following URLs for information on how to access a public time source:

      `http://www.eecis.udel.edu/~mills/ntp/servers.html`
      `http://www.boulder.nist.gov/timefreq/general/pdf/1383.pdf`

   d. **Close the file** `/etc/inet/ntp.conf`**.**

e. Start the `xntpd`(1M) daemon.

```
# /etc/init.d/xntpd start
```

Editing `mcf` Files on Other Hosts

Perform the tasks described in this section if you are configuring one of the following types of file systems:

- Sun StorEdge QFS highly available file system in a Sun Cluster environment
- Sun StorEdge QFS shared file system on the Solaris OS
- Sun StorEdge QFS shared file system in a Sun Cluster environment

The lines that define a particular file system must be identical in the `mcf` files on all host systems that support the file system. Only one `mcf` file can reside on a host. Because you can have other, additional Sun StorEdge QFS file systems defined in an `mcf` file, the `mcf` files on different hosts might not be identical.

Note – If you update a metadata server's `mcf` file at any time after the Sun StorEdge QFS shared file system is mounted, make sure that you update the `mcf` files as necessary on all hosts that can access that shared file system.

▼ To Edit `mcf` Files for a Highly Available File System in a Sun Cluster Environment

Perform this procedure for a Sun StorEdge QFS highly available file system in a Sun Cluster environment, on each host that you want to have support the file system you are configuring.

1. **Log in to the Sun Cluster node.**
2. **Become superuser.**
3. **Use `vi`(1) or another editor to create an `mcf` file on that node.**

   If an `mcf` file already exists on the host, add the lines for the new file system to this `mcf` file.
4. **Copy the lines that define the file system from the primary node's `mcf` file to this node's `mcf` file.**

▼ To Edit `mcf` Files for a Sun StorEdge QFS Shared File System

Perform this procedure for each host that you want to include in a shared file system in a Solaris or Sun Cluster environment.

1. **Log in to the host.**
2. **Become superuser.**
3. **Use the `format`(1M) command to verify the presence of client host disks.**
4. **Use `vi`(1) or another editor to create an `mcf` file.**

   If an `mcf` file already exists on the host, add the lines for the new file system to this `mcf` file.

5. **Issue the `samfsconfig`(1M) command.**

   Examine this command's output to locate the local device names for each additional host to be configured in the Sun StorEdge QFS shared file system.

   The `samfsconfig`(1M) command generates configuration information that can help you to identify the devices included in the Sun StorEdge 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.

6. **Update the `mcf` file on other client hosts.**

   To access or mount a shared file system, a host system must have that file system defined in its `mcf` file. The content of `mcf` files varies, depending on whether the Solaris OS or Sun Cluster environment hosts the file system, as follows:

   - There are three types of Solaris hosts: the metadata server, clients that are potential metadata servers, and clients that can never be metadata servers. For clients that can never become metadata servers, use the keyword `nodev` in the Equipment Identifier field. The examples following this procedure show how to use this keyword.
   - There are two types of Sun Cluster hosts: the primary metadata server and potential metadata servers. There are no hosts that cannot be metadata servers because the Sun Cluster software fails over system resources in the event of a node failure.

   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 Sun StorEdge 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 fields 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.

Examples

Example 1 – Solaris OS hosts. CODE EXAMPLE 4-1 shows how the `samfsconfig`(1M) command is used to retrieve device information for family set `sharefs1` on client `tethys`. Because `tethys` is a potential metadata server, it is connected to the same metadata disks as `titan`.

CODE EXAMPLE 4-1 `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
/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. 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.

CODE EXAMPLE 4-2 shows the resulting `mcf` file.

CODE EXAMPLE 4-2 `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 CODE EXAMPLE 4-2, the Equipment Ordinal numbers match those of the `mcf` file for metadata server `titan` (see "Configuration Example for a Shared File System on a Solaris OS Platform" on page 118). These Equipment Ordinal numbers must not already be in use on client host `tethys` or any other client host.

Example 2 – Solaris OS hosts. CODE EXAMPLE 4-3 shows how the `samfsconfig`(1M) command is used to retrieve device information for family set `sharefs1` on client host `mimas`. Because `mimas` can never become a metadata server, it is not connected to the metadata disks.

CODE EXAMPLE 4-3 `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
# /dev/dsk/c1t50020F2300005D22d0s6   12    mr   sharefs1   on
# /dev/dsk/c1t50020F2300006099d0s6   13    mr   sharefs1   on
# /dev/dsk/c1t50020F230000651Cd0s6   14    mr   sharefs1   on
```

In the command output, 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. 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 field. 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.

CODE EXAMPLE 4-4 shows the resulting `mcf` file.

CODE EXAMPLE 4-4 `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
```

Creating the Shared Hosts File

Perform the tasks described in this section if you are configuring one of the following types of file systems:

- Sun StorEdge QFS shared file system on the Solaris OS
- Sun StorEdge QFS shared file system in a Sun Cluster environment

---

Note – Alternatively, you can set up a shared hosts file using File System Manager. For instructions, see the "Planning a Shared File System" topic in the File System Manager online help.

▼ To Create the Shared Hosts File on the Metadata Server

When you create a shared file system, the system copies information from the hosts file to the shared hosts file on the metadata server. You update this information when you issue the `samsharefs`(1M) `-u` command.

1. **Use the `cd`(1) command to change to directory `/etc/opt/SUNWsamfs`.**
2. **Use `vi`(1) or another editor to create an ASCII hosts file called `hosts.`*fs-name*.**

   For *fs-name*, specify the Family Set name of the Sun StorEdge QFS shared file system.

   Comments are permitted in the hosts file. Comment lines must begin with a pound character (#). Characters to the right of the pound character are ignored.
3. **Use the information in** TABLE 4-1 **to fill in the lines of the hosts file.**

   File `hosts.`*fs-name* contains configuration information pertaining to all hosts in the Sun StorEdge QFS shared file system. The ASCII hosts file defines the hosts that can share the Family Set name.

   TABLE 4-1 shows the fields in the hosts file.

TABLE 4-1 Hosts File Fields

Field Number	Content
1	The Host Name field. This field must contain an alphanumeric host name. It defines the Sun StorEdge QFS shared file system hosts. You can use the output from the `hostname`(1) command to create this field.

TABLE 4-1 Hosts File Fields *(Continued)*

Field Number	Content
2	The Host IP Addresses field. This field must contain a comma-separated list of host IP addresses. You can use the output from the `ifconfig`(1M) `-a` command to create this field. You can specify the individual addresses in one of the following ways: • Dotted-decimal IP address form • IP version 6 hexadecimal address form • A symbolic name that the local domain name service (DNS) can resolve to a particular host interface The metadata server uses this field to determine whether a host is allowed to connect to the Sun StorEdge QFS shared file system. If the metadata server receives a connection attempt from any interface not listed in this field, it rejects the connection attempt. Conversely, use care when adding elements here, because the metadata server accepts any host with an IP address that matches an address in this field. Client hosts use this field to determine the metadata server interfaces to use when attempting to connect to the metadata server. A host evaluates the addresses from left to right and attempts a connection using the first responding address in the list.
3	The Server field. This field must contain either a dash character (`-`) or an integer ranging from `0` through *n*. The `-` and the `0` are equivalent. If the value of the Server field is a nonzero integer, the host is a potential metadata server. The rest of the row defines the server as a metadata host. The metadata server processes all the metadata modification for the file system. At any one time there is at most one metadata server host, and that metadata server supports archiving, staging, releasing, and recycling for a Sun StorEdge QFS shared file system. If the Server field is `-` or `0`, the host is not eligible to be a metadata server.
4	Reserved for future use by Sun Microsystems. This field must contain either a dash character (`-`) or a `0`. The `-` and the `0` are equivalent.
5	The Server Host field. This field can contain either a blank or the `server` keyword in the row that defines the active metadata server. Only one row in the hosts file can contain the `server` keyword. This field must be blank in all other rows.

The system reads and manipulates the hosts file. You can use the `samsharefs`(1M) command to examine metadata server and client host information about a running system.

Example for Solaris OS Hosts

CODE EXAMPLE 4-5 is an example hosts file that shows four hosts.

CODE EXAMPLE 4-5 Sun StorEdge 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                   -        -
```

This hosts file contains fields of information and comment lines for the `sharefs1` file system. In this example, the number 1 in the Server Priority field defines `titan` as the primary metadata server. If `titan` is unavailable, the next metadata server is `tethys`, as indicated by the number `2` in this field. Note that neither `mimas` nor `dione` can ever be a metadata server.

Example for Sun Cluster Hosts

If you are configuring a Sun StorEdge QFS shared file system in a Sun Cluster environment, every host is a potential metadata server. The hosts files and the local hosts configuration files must contain node names in the Host Names field and Sun Cluster private interconnect names in the Host IP Addresses field.

CODE EXAMPLE 4-6 shows the local hosts configuration file for shared file system `sharefs1`. This file system's participating hosts are Sun Cluster nodes `scnode-A` and `scnode-B`. Each node's private interconnect name is listed in the Host IP Addresses field.

CODE EXAMPLE 4-6 Sun StorEdge QFS Shared File System Hosts File Example

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host   Host IP                              Server   Not  Server
# Name   Addresses                           Priority Used Host
# ----   --------------------------------- -------- ---- -----
scnode-A clusternode1-priv                   1        -    server
scnode-B clusternode2-priv                   2        -
```

▼ To Create the Local Hosts File on a Client

Perform this procedure under the following circumstances:

- If your Sun StorEdge QFS shared file system host systems have multiple host interfaces. You can use this file to specify how file system traffic should flow over public and private networks in your environment.
- If you are configuring a Sun StorEdge QFS shared file system on Solaris OS hosts. Do not create this file if you are configuring a Sun StorEdge QFS shared file system in a Sun Cluster environment.

Follow these steps for each client host that you want to include in the Sun StorEdge QFS shared file system.

1. **Create the local hosts configuration file on the client host.**

   Using `vi`(1) or another editor, create an ASCII local hosts configuration file that defines the host interfaces that the metadata server and the client hosts can use when accessing the file system. 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 Sun StorEdge QFS shared file system.

   Comments are permitted in the local host configuration file. Comment lines must begin with a pound character (#). Characters to the right of the pound character are ignored. TABLE 4-2 shows the fields in the local hosts configuration file.

TABLE 4-2 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 Sun StorEdge QFS shared file system.
2	The Host Interfaces field. This field must contain a comma-separated list of host interface addresses. You can use the output from the `ifconfig`(1M) `-a` command to create this field. You can specify the individual interfaces in one of the following ways: • Dotted-decimal IP address form • IP version 6 hexadecimal address form • A symbolic name that the local domain name service (DNS) can resolve to a particular host interface The client hosts use this field to determine the metadata server interfaces to use when attempting to connect to the metadata server. The system evaluates the addresses from left to right and attempts a connection using the first responding address in the list that is also included in the shared hosts file.

How Metadata Server Addresses Are Obtained

The information in this section might be useful when you are debugging.

In a Sun StorEdge QFS shared file system, each client host obtains the list of metadata server IP addresses from the shared hosts file.

The metadata server and the client hosts use the shared hosts 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 metadata server. This process is as follows:

Note – The term *client*, as in *network client*, is used to refer to both client hosts and the metadata server host.

---

1. The client obtains the list of metadata server host IP interfaces from the file system's on-disk shared hosts file. To examine this file, issue the `samsharefs`(1M) command from the metadata server or from a potential metadata server.
2. The client searches for an `/etc/opt/SUNWsamfs/hosts.`*fsname*`.local` file. Depending on the outcome of the search, one of the following occurs:
   - If a `hosts.`*fsname*`.local` file does not exist, the client attempts to connect, in turn, to each address in the server's line in the shared hosts file until it succeeds in connecting.
   - If the `hosts.`*fsname*`.local` file exists, the client performs the following tasks:
     i. It compares the list of addresses for the metadata server from both the shared hosts file on the file system and the `hosts.`*fsname*`.local` file.
     ii. 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 FIGURE D-1 in Appendix D. CODE EXAMPLE 4-5 on page 65 shows the hosts file for this configuration. FIGURE 4-1 shows the interfaces to these systems.

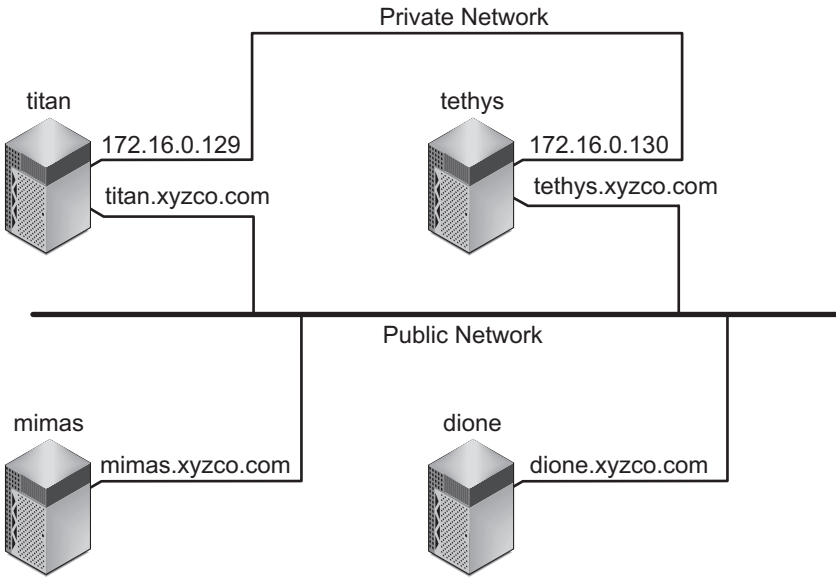

FIGURE 4-1 Network Interfaces

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. CODE EXAMPLE 4-7 shows the information in these files.

CODE EXAMPLE 4-7 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`' public interfaces, and never attempt to connect to `titan`'s or `tethys`' unreachable private interfaces, the

system administrator has created identical copies of `/etc/opt/SUNWsamfs/hosts.sharefs1.local` on `mimas` and `dione`. CODE EXAMPLE 4-8 shows the information in these files.

CODE EXAMPLE 4-8 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
```

Verifying That the Daemons Are Running

Perform the tasks described in this section if you are configuring the following types of file systems:

- Sun StorEdge QFS shared file system on a Solaris OS
- Sun StorEdge QFS shared file system in a Sun Cluster environment

▼ To Verify the Daemons

Perform these steps on each host that can mount the file system.

1. **Use the `ps`(1) and `grep`(1) commands to determine whether the `sam-sharefsd` daemon is running for this file system.**

   For example:

```
# 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
```

   This example shows that the `sam-sharefsd` daemon is active for the `sharefs1` file system.

Note – If the `sam-sharefsd` daemon is active for your Sun StorEdge QFS shared file system, you need to perform some diagnostic procedures. For information about these procedures, see the *Sun StorEdge QFS Configuration and Administration Guide*.

2. **If the output from this command indicates that the `sam-sharefsd` daemon is not running, determine whether the `sam-fsd` daemon is running as follows:**

   a. **Use the `ps`(1) and `grep`(1) commands to verify that the `sam-fsd` daemon is running for this file system.**

   b. **Examine the output.**

      CODE EXAMPLE 4-9 shows `sam-fsd` output that indicates that the daemon is running.

CODE EXAMPLE 4-9 `sam-fsd`(1M) Output That Shows the `sam-fsd` Daemon is Running

```
cur% ps -ef | grep sam-fsd
 user1 16435 16314  0 16:52:36 pts/13   0:00 grep sam-fsd
    root   679     1  0   Aug 24 ?        0:00
/usr/lib/fs/samfs/sam-fsd
```

3. **Do one of the following:**

   - If the output indicates that the `sam-fsd` daemon is not running, and if no file system has been accessed since the system's last boot, issue the `samd`(1M) `config` command, as follows:

```
# samd config
```

   - If the output indicates that the `sam-fsd` daemon is running, enable tracing in the `defaults.conf`(4) file and check the following files to determine whether configuration errors are causing the problem:
     - `/var/opt/SUNWsamfs/trace/sam-fsd`
     - `/var/opt/SUNWsamfs/trace/sam-sharefsd`

Configuring the `SUNW.qfs` Resource Type

Perform the task described in this section if you are configuring a Sun StorEdge QFS shared file system on a Sun Cluster platform.

▼ To Enable a Sun StorEdge QFS Shared File System as a SUNW.qfs(5) Resource

1. **Log in to the metadata server as superuser.**
2. **Use the** scrgadm**(1M)** -p **command and search for the** SUNW.qfs**(5) resource type.**

   For example:

```
metadataserver# scrgadm -p | grep SUNW.qfs
```

3. **If the** SUNW.qfs **resource type is missing, issue the following command:**

```
metadataserver# scrgadm -a -t SUNW.qfs
```

4. **Use the** scrgadm**(1M) command to set the** FilesystemCheckCommand **property of the** SUNW.qfs**(5) resource type to** /bin/true**.**

   The SUNW.qfs(5) resource type is part of the Sun StorEdge QFS software package. Configuring the resource type for use with your shared file system makes the shared file system's metadata server highly available. Sun Cluster scalable applications can then access data contained in the file system. For more information, see the *Sun StorEdge QFS Configuration and Administration Guide*.

   CODE EXAMPLE 4-10 shows how to use the scrgadm(1M) command to register and configure the SUNW.qfs resource type. In this example, the nodes are scnode-A and scnode-B. /global/sharefs1 is the mount point as specified in the /etc/vfstab file.

CODE EXAMPLE 4-10 Configuring a 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/sharefs1
```

Configuring the HA Storage Plus Resource

Perform the task in this section if you are configuring a Sun StorEdge QFS highly available file system on a Sun Cluster platform.

▼ To Configure a Highly Available File System as an HA Storage Plus Resource

- **Use the** `scrgadm`**(1M) command to set the** `FilesystemCheckCommand` **property of HA Storage Plus to** `/bin/true`**.**

  All other resource properties for HA Storage Plus apply as specified in `SUNW.HAStoragePlus`(5).

  The following example command shows how to use the `scrgadm`(1M) command to configure an HA Storage Plus resource:

```
# scrgadm -a -g qfs-rg -j ha-qfs -t SUNW.HAStoragePlus \
        -x FilesystemMountPoints=/global/qfs1 \
        -x FilesystemCheckCommand=/bin/true
```

Bringing the Shared Resource Online

Perform the task described in this section if you are configuring the following types of file systems:

- Sun StorEdge QFS shared file system in a Sun Cluster environment
- Sun StorEdge QFS highly available file system in a Sun Cluster environment

▼ To Bring the Shared Resource Online

1. **Verify that the file system is mounted on all nodes.**

   If it is not mounted, go back to "Mounting the File System" on page 42 and follow the instructions there.

2. **Log in to the appropriate host.**
   - If you are configuring a Sun StorEdge QFS shared file system, log in to the metadata server.
   - If you are configuring a Sun StorEdge QFS highly available file system, log in to the node upon which the file system is based.

3. **Use the `scswitch`(1M) command to move the file system resource to another node.**

   For example:

```
metadataserver# scswitch -Z -g qfs-rg
```

4. **Use the `scstat`(1M) command to verify that the file system resource was moved successfully.**

   For example:

```
metadataserver# scstat
< information deleted from this output >
-- Resources --
Resource Name    Node Name  State     Status Message
-------------    ---------  -----     --------------
Resource: qfs-res   ash     Online    Online
Resource: qfs-res   elm     Offline   Offline
Resource: qfs-res   oak     Offline   Offline
```

Verifying the Resource Group on All Nodes

Perform the task described in this section if you are configuring the following types of file systems:

- Sun StorEdge QFS shared file system in a Sun Cluster environment. This step ensures that the metadata server can move from node to node.

- Sun StorEdge QFS highly available file system in a Sun Cluster environment. This step ensures that the file system can move from node to node when the Sun Cluster software performs a failover.

▼ To Verify the Resource Group on All Nodes

Perform these steps for each node in the cluster, with a final return to the original server.

1. **From any node in the Sun Cluster environment, use the `scswitch`(1M) command to move the file system resource from one node to another.**

   For example:

```
server# scswitch -z -g qfs-rg -h elm
```

2. **Use the `scstat`(1M) command to verify that the file system resource was moved successfully.**

   For example:

```
server# scstat
-- Resources --
Resource Name    Node Name  State     Status Message
-------------    ---------  -----     --------------
Resource: qfs-res   ash     Offline   Offline
Resource: qfs-res   elm     Online    Online
Resource: qfs-res   oak     Offline   Offline
```

CHAPTER 5

Upgrade and Configuration Tasks

This chapter describes the procedures for upgrading a server to a new release of the Sun StorEdge software. Use these procedures if you are upgrading your Sun StorEdge QFS file system. You must perform all the tasks in this chapter as superuser.

This chapter contains the following sections:

- "Preparing for an Upgrade" on page 75
- "Removing Existing Sun StorEdge QFS Software" on page 87
- "Adding the Upgrade Packages" on page 88
- "Installing File System Manager" on page 89
- "Restoring the File System" on page 91
- "Upgrading the Solaris OS" on page 95

Preparing for an Upgrade

Follow the instructions in this section to prepare for a Sun StorEdge QFS upgrade.

Upgrade Considerations

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

- It is wise to move to the new host while the existing host is still in operation. This enables you to install, configure, and test the new hardware platform with your applications.

- Moving to a new host system is equivalent to installing the Sun StorEdge QFS software for the first time. In SAM-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.
- Before powering down the old host system, decide whether the backup copies you have on hand are sufficient. You might want new dump files to re-create the file system on the new server. For more information about creating a dump file, see "Setting Up Dump Files" on page 54.

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 you make these changes.

The `samfsconfig`(1M) command examines the devices you specify, determines whether any of them have Sun StorEdge QFS superblocks on them, and writes this information to `stdout`. 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 re-create a damaged, missing, or incorrect `mcf` file.

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 Sun StorEdge 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.

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 were for devices that were not accounted for in the environment and that the system administrator therefore wanted to examine for Sun StorEdge QFS family sets. The results displayed in CODE EXAMPLE 5-1 show some old fragments of family sets and several complete instances.

CODE EXAMPLE 5-1 Example 1 - Output From `samfsconfig`(1M) Command

```
mn# samfsconfig -v 'cat /tmp/dev_files'
Device '/dev/dsk/c5t10d0s0' has a SAM-FS superblock.
Couldn't open '/dev/dsk/c5t10d0s1': I/O error
Device '/dev/dsk/c5t10d0s3' has a SAM-FS superblock.
Device '/dev/dsk/c5t10d0s4' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t10d0s5' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t10d0s6' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t10d0s7' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t11d0s0' has a SAM-FS superblock.
Couldn't open '/dev/dsk/c5t11d0s1': I/O error
Device '/dev/dsk/c5t11d0s3' has a SAM-FS superblock.
Device '/dev/dsk/c5t11d0s4' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t11d0s5' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t11d0s6' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t11d0s7' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t12d0s0' has a SAM-FS superblock.
Couldn't open '/dev/dsk/c5t12d0s1': I/O error
Device '/dev/dsk/c5t12d0s3' has a SAM-FS superblock.
Device '/dev/dsk/c5t12d0s4' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t12d0s5' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t12d0s6' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t12d0s7' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t13d0s0' has a SAM-FS superblock.
Couldn't open '/dev/dsk/c5t13d0s1': I/O error
Device '/dev/dsk/c5t13d0s3' has a SAM-FS superblock.
Device '/dev/dsk/c5t13d0s4' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t13d0s5' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t13d0s6' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t13d0s7' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t8d0s0' has a SAM-FS superblock.
Device '/dev/dsk/c5t8d0s1' has a SAM-FS superblock.
Device '/dev/dsk/c5t8d0s3' has a SAM-FS superblock.
Device '/dev/dsk/c5t8d0s4' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t8d0s5' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t8d0s6' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t8d0s7' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t9d0s0' has a SAM-FS superblock.
Couldn't open '/dev/dsk/c5t9d0s1': I/O error
Device '/dev/dsk/c5t9d0s3' has a SAM-FS superblock.
```

CODE EXAMPLE 5-1 Example 1 - Output From `samfsconfig`(1M) Command *(Continued)*

```
Device '/dev/dsk/c5t9d0s4' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t9d0s5' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t9d0s6' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t9d0s7' doesn't have a SAM-FS superblock (SBLK).
13 SAM-FS devices found.
#
# Family Set 'qfs1' Created Mon Jun 25 10:37:52 2004
#
# Missing slices
# Ordinal 0
# /dev/dsk/c5t8d0s1 10 mm qfs1 -
#
# Family Set 'qfs1' Created Wed Jul 11 08:47:38 2004
#
qfs1 200 ma qfs1 - shared
/dev/dsk/c5t8d0s3  201   mm    qfs1  -
/dev/dsk/c5t9d0s3  202   mr    qfs1  -
/dev/dsk/c5t10d0s3 203   mr    qfs1  -
/dev/dsk/c5t11d0s3 204   mr    qfs1  -
/dev/dsk/c5t12d0s3 205   mr    qfs1  -
/dev/dsk/c5t13d0s3 206   mr    qfs1  -
#
# Family Set 'sqfs1' Created Wed Nov  7 16:55:19 2004
#
sqfs1 100 ma sqfs1 - shared
/dev/dsk/c5t8d0s0  101   mm    sqfs1  -
/dev/dsk/c5t9d0s0  102   mr    sqfs1  -
/dev/dsk/c5t10d0s0 103   g0    sqfs1  -
/dev/dsk/c5t11d0s0 104   g0    sqfs1  -
/dev/dsk/c5t12d0s0 105   g1    sqfs1  -
/dev/dsk/c5t13d0s0 106   g1    sqfs1  -
#
```

Example 2

In the output shown in CODE EXAMPLE 5-2, 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.

CODE EXAMPLE 5-2 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
```

CODE EXAMPLE 5-2 Example 2 - Output from `samfsconfig` Command *(Continued)*

```
> /dev/dsk/c3t50020F23000055A8d0s2    161    mm   shsam1  -
> /dev/dsk/c3t50020F23000055A8d0s0    161    mm   shsam1  -
/dev/dsk/c3t50020F23000055A8d0s1    162    mr   shsam1  -
> /dev/dsk/c3t50020F23000078F1d0s0    163    mr   shsam1  -
> /dev/dsk/c3t50020F23000078F1d0s2    163    mr   shsam1  -
/dev/dsk/c3t50020F23000078F1d0s1    164    mr   shsam1  -
```

Preparing for a Hardware Device Upgrade

This section prepares you for hardware upgrades to devices within your environment.

General Prerequisites

Before starting the upgrade process, be sure to do the following:

- Determine whether the hardware addition or change requires a software upgrade from Sun Microsystems.

  Examples of changes that do not require a software upgrade include additions to memory and increases in disk cache. Examples of changes that require a software upgrade include changes to the class of your server or significant increases in storage capacity.

- If you are switching from a SPARC to an AMD server platform (or from AMD to SPARC), you must take precautions to prevent loss of data. See "Switching Between SPARC and AMD Platforms" on page 80 for details.
- 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 Ordinal values in your old and new `mcf` files. For information about the `mcf` file, see the `mcf`(4) man page.
- Decide whether the backup copies you have on hand are sufficient. For information about backing up your data and metadata, see the procedures described in "Setting Up Dump Files" on page 54.
  - In a Sun StorEdge QFS environment, the `qfsdump`(1M) command dumps all data and metadata. For more information about this process, see the `qfsdump`(1M) man page.

- In SAM-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 SAM-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 SAM-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.

Switching Between SPARC and AMD Platforms

Support for x86 hardware platforms was added in the 4U4 version of the software. The following are some important considerations if you are combining or changing between SPARC and x86 hardware platforms:

- Sun StorEdge QFS software is supported only for the Solaris 10 OS on x64 platforms (AMD64 architecture), not for the EM64T architecture. With the exception of the Sun StorEdge QFS shared Linux client, it is also not supported for any 32-bit x86 architectures.
- All functionality that is supported by Sun StorEdge QFS software on the SPARC platform is also supported on the x64 platform except for the following:
  - Sun StorEdge QFS software for the Solaris 10 OS on x64 platforms does not support Sun Cluster software.
  - The ADIC/Grau, Fujitsu LMF, IBM 3494, and Sony network-attached libraries are not supported on x64 platforms. StorageTek (STK) ACSLS-attached automated libraries are supported on x64 platforms.
  - Optical (MO and UDO) storage libraries and drives are not supported on x64 platforms.

    - SANergy software is not supported on x64 platforms.
    - SCSI-attached tape drives are not supported on x64 platforms because of a lack of support in the SCSI HBA 64-bit drivers for large block sizes. Both SCSI-attached libraries and fibre-attached libraries are supported with fibre-attached tape drives.
- EFI labels are required on all disks if your Sun StorEdge QFS shared file system configuration contains both the Solaris 10 OS on x64 platforms and the Solaris 9 or Solaris 10 OS on SPARC platforms. See "Configuring EFI Labels for Shared x64 and SPARC Volumes" on page 82 for information on relabeling disks.
- You can add Sun StorEdge QFS shared file system Linux clients to Solaris SPARC configurations that are using SMI VTOC8 disk labels and to Solaris AMD64 configurations that are using SMI VTOC16 disk labels. You can also add Sun StorEdge QFS shared file system Linux clients to these configurations when they are using EFI disk labels, but you may need to rebuild the Linux kernel for this capability. This is dependent on the particular Linux distribution. See the Linux client README file for more details.
- You must exercise caution when accessing the same SAN attached storage from a Solaris environment on both SPARC and x64 platforms. The Solaris OS on x64 platforms cannot interpret the SMI VTOC8 disk label created by the Solaris OS on SPARC platforms, and the Solaris OS on SPARC platforms cannot interpret the SMI VTOC16 disk label created by the Solaris OS on x64. This can make it appear as though a disk is unlabeled, when in fact it is labeled and in use by a platform of a different architecture type. For example, a disk that is labeled with SMI VTOC8 may have mounted partitions in use by Solaris on a SPARC platform, but will appear as unlabeled when viewed with the `format`(1M) partition command by Solaris on an x64 platform. If you make the mistake of running `fdisk`(1M) as prompted by the `format`(1M) command, you will destroy the contents of that disk.
- You cannot change the architecture type of the server responsible for control of the file system metadata operations (i.e. the server that was used to create the file system with the `sammkfs`(1M) command). For a Sun StorEdge QFS standalone file system, this means that you cannot mount the file system on a server that has a different architecture type from the one that created it. For a Sun StorEdge QFS shared file system, this means that you cannot change the architecture type of the metadata server or any potential metadata servers. This is because the different architectures use different byte-ordering schemes (endianness). However, you can migrate data from one architecture type to the other by copying the file system to temporary storage using either `qfsdump`(1M) or `samfsdump`(1M), re-creating the file system using `sammkfs`(1M), and then repopulating the file system with `qfsrestore`(1M) or `samfsrestore`(1M).
- The Sun StorEdge Traffic Manager I/O multipathing feature (MPxIO) is disabled by default for the Solaris 9 and 10 OS on the SPARC platform and enabled by default for the Solaris 10 OS on x64. This feature should be configured the same

way for all systems in your Sun StorEdge QFS shared file system configuration. It is configured in `/kernel/drv/scsi_vhci.conf` for the Solaris 9 OS and in `/kernel/drv/fp.conf` for the Solaris 10 OS.

- In a Sun StorEdge QFS shared file system environment, a configuration error will be generated if you have potential metadata servers of different architecture types (SPARC and x64) defined in the `/etc/opt/SUNWsamfs/hosts.fs` file.

Configuring EFI Labels for Shared x64 and SPARC Volumes

Caution – Relabeling a disk will destroy the contents of that disk.

Use the Solaris `prtvtoc`(1M) command to determine whether a disk contains SMI or EFI labels. Under the Dimensions section of the output, SMI labels list the number of accessible cylinders, whereas EFI labels list the number of accessible sectors.

To convert disk labels from the default SMI VTOC8 to EFI, copy the file system to temporary storage using `qfsdump`(1M) or `samfsdump`(1M), relabel the disks with EFI labels using the `format -e` command, re-create the file system using `sammkfs`(1M), and repopulate the file system with `qfsrestore`(1M) or `samfsrestore`(1M).

When using the Solaris `format -e` command to create EFI labels, you can select the partition command from the menu to create and modify partitions (slices). When doing this, you must specify a tag id name of `usr`, rather than `stand` or `unassigned`, for EFI labels.

Note that EFI labels reserve the first 34 sectors, which misaligns Sun RAID-5 storage from a performance perspective. Unless you realign the storage, you will incur a RAID-5 read/modify/write performance penalty whenever writing. You can avoid this performance penalty by selecting the proper starting sector for all disk partitions for your particular storage configuration. For example, an 8+P Sun StorEdge T3 array with a 64K block size should have starting sectors that are multiples of 1024 for all disk slices ((8 * 64 * 1024) / 512 = 1024). Similarly, a 5+P Sun StorEdge 3510 FC array with a 128K block size should have starting sectors that are multiples of 1280 for all disk slices ((5 * 128 * 1024) / 512 = 1280).

Backing Up Existing File Systems

Back up your existing file systems if the following conditions exist:

- You are currently using a version 1 superblock with a Sun StorEdge QFS 4U0 system and you want to reinitialize your file systems with a version 2 superblock. In "To Reinitialize and Restore the File System" on page 93, you reinitialize the file systems and restore your data.
- You suspect that your current `qfsdump`(1M) file is incorrect or outdated.

The following subsections explain the differences between these two superblocks and present the procedure for backing up your file systems:

- "Using the Version 1 and Version 2 Superblocks" on page 83
- "To Back Up a File System" on page 84

CODE EXAMPLE 5-3 shows the `samfsinfo`(1M) command you use to retrieve information about the `qfs2` file system. The second line of output indicates that this file system is using a version 2 superblock.

CODE EXAMPLE 5-3 Using `samfsinfo`(1M)

```
# samfsinfo qfs2
samfsinfo: filesystem qfs2 is mounted.
name: qfs2       version:     2    shared
time:     Sun Sep 28 08:20:11 2003
count:    3
capacity:       05aa8000          DAU:         64
space:          0405ba00
meta capacity: 00b4bd20           meta DAU:    16
meta space:     00b054c0
ord  eq   capacity      space   device
  0  21   00b4bd20   00b054c0   /dev/md/dsk/d0
  1  22   02d54000   01f43d80   /dev/dsk/c9t50020F2300010D6Cd0s6
  2  23   02d54000   02117c80   /dev/dsk/c9t50020F2300010570d0s6
```

Using the Version 1 and Version 2 Superblocks

Sun StorEdge QFS 4U1 and later releases support both a version 1 superblock and a version 2 superblock. Only the version 2 superblock supports the following features:

- Access Control Lists (ACLs)
- Sun StorEdge QFS shared file system
- `md` devices in Sun StorEdge QFS or SAM-QFS (`ma`) file systems
- Dual-sized disk allocation units (DAUs) on `mm` devices

The Sun StorEdge QFS 4U1 and later releases support both the version 1 and version 2 superblocks. You can use the `sammkfs`(1M) command to create a version 2 superblock, but you cannot initialize any file systems with version 1 superblocks. In addition, it is not possible to move files from a file system with a version 2 superblock back to a file system with a version 1 superblock.

After you reinitialize a file system, you can use the `qfsrestore`(1M) command to restore files to the new file system from the dump file created during the backup process.

If you are upgrading from a Sun QFS 4U0 system, note that the Sun StorEdge QFS 4U0 file system allows you to initialize file systems with either a version 1 or a version 2 superblock. If you want to reinitialize any of the file systems that have a version 1 superblock and remake them with a version 2 superblock, back up these file systems now.

Note – Sun StorEdge QFS 4U2 and later releases do not enable you to initialize a file system with a version 1 superblock. These more recent versions enable you to initialize file systems only with the version 2 superblock.

▼ To Back Up a File System

Follow these steps for each Sun StorEdge QFS file system in your environment.

1. **Become superuser from a console connection.**

   If you have not already logged in as `root`, do so now.

2. **Use the `boot`(1M) command to boot the system in single-user mode:**

```
# boot -s
```

3. **Use the `mount`(1M) command to mount the Sun StorEdge QFS file system.**

   For example:

```
# mount /qfs1
```

4. **Use the `qfsdump`(1M) command to back up the file data and metadata of the Sun StorEdge QFS file system.**

   The `qfsdump`(1M) command dumps file names, inode information, and file data. The destination of the `qfsdump`(1M) output (generally a file) must be as large as or larger than the Sun StorEdge QFS file system that you are backing up. The destination location (disk or tape) must have enough space to hold the amount of

file data and metadata that you are dumping. For more information about using the qfsdump(1M) command, see "Setting Up Dump Files" on page 54 or see the qfsdump(1M) man page.

Dump each file system to a location outside of the Sun StorEdge QFS file system. For more information, see the qfsdump(1M) man page.

For example, if you have a file system named qfs1 (mounted at /qfs1) that you want to back up, your choices are as follows:

- You can write the qfsdump(1M) output to a tape device.

  CODE EXAMPLE 5-4 shows how to write to a tape in device /dev/rmt/1cbn.

CODE EXAMPLE 5-4 Writing qfsdump(1M) Output to a Tape Device

```
# cd /qfs1
# qfsdump -f /dev/rmt/1cbn
```

- You can write the qfsdump(1M) output to a file in a UFS file system.

  CODE EXAMPLE 5-5 shows how to write to a file in a UFS file system.

CODE EXAMPLE 5-5 Writing qfsdump(1M) Output to a File in the UFS File System

```
# cd /qfs1
# qfsdump -f /save/qfs/qfs1.bak
```

- You can initialize a new Sun StorEdge QFS file system, using a Sun StorEdge QFS 4U2 or later release, and perform the qfsrestore(1M) command directly in that new Sun StorEdge QFS file system.

  This alternative is applicable only if you already have the Sun StorEdge QFS software installed and operational as a file system somewhere in your environment. This alternative also assumes that you want to use the features supported by the Sun StorEdge QFS 4U2 or later release and the version 2 superblock.

  For example, assume that you want to write the dump file into a second Sun StorEdge QFS file system called qfs2 (mounted at /qfs2) and that you initialized the qfs2 file system using Sun StorEdge QFS 4U2 or later software. CODE EXAMPLE 5-6 shows how to accomplish this using commands.

CODE EXAMPLE 5-6 Writing qfsdump(1M) Output to a Sun StorEdge QFS File System

```
# mount /qfs2
# cd /qfs1
# qfsdump -f - | (cd /qfs2; qfsrestore -f -)
```

For more information about backing up your file systems, see "Setting Up Dump Files" on page 54.

▼ To Unshare File Systems

Perform this task if your Sun StorEdge QFS file systems are NFS shared file systems.

- **Use the `unshare`(1M) command on the Sun StorEdge QFS file system.**

  For example, the following command unshares the `qfs1` file system:

```
# unshare /qfs1
```

Unmounting File Systems

You can unmount a file system using any of the following methods described in this section. After the file system is unmounted, you can proceed to "Removing Existing Sun StorEdge QFS Software" on page 87.

Note – To unmount a Sun StorEdge QFS shared file system, follow the instructions in the *Sun StorEdge QFS Configuration and Administration Guide*.

▼ To Unmount Using File System Manager

1. **From the Servers page, click the name of the server on which the file system is located.**

   The File System Summary page is displayed.

2. **Select the radio button next to the file system that you want to unmount.**

3. **From the Operations menu, choose Unmount.**

▼ To Unmount Using CLI Commands

- **Use the `umount`(1M) command to unmount each Sun StorEdge QFS file system.**

  If necessary, use the `-f` option to the `umount`(1M) command. The `-f` option forces a file system to unmount.

  If `umount`(1M) is not successful, it might be because files in the file system are being used or because you have used the `cd` command to change to a directory that is within the file system. In this case, follow these steps:

1. **Use the `fuser`(1M) command to determine whether any processes are still busy.**

   For example, the following command queries the `qfs1` file system:

```
# fuser -uc /qfs1
```

2. **If any processes are still busy, use the `kill`(1M) command to terminate them.**
3. **Use the `umount`(1M) command to unmount each Sun StorEdge QFS file system.**

▼ To Unmount by Editing the `/etc/vfstab` File and Rebooting

1. **Edit the `/etc/vfstab` file.**

   For all Sun StorEdge QFS file systems, change the Mount at Boot field from `yes` or `delay` to `no`.

2. **Reboot the system.**

Removing Existing Sun StorEdge QFS Software

Use the `pkgrm`(1M) command to remove the existing software. You must remove the existing Sun StorEdge QFS package before installing the new package.

If you are using any optional Sun StorEdge QFS packages, you should make sure that you remove these packages before removing the main `SUNWqfs` packages. The installation script prompts you to confirm several of the removal steps.

▼ To Remove Existing Software

1. **Use the `pkginfo`(1) command to determine which Sun StorEdge QFS software packages are installed on your system.**

   For example:

```
# pkginfo | grep qfs
```

2. **Use the `pkgrm`(1M) command to remove the existing Sun StorEdge QFS software.**

   The following example command removes the `SUNWqfsu` and the `SUNWqfsr` packages from a 4U1 release:

```
# pkgrm SUNWqfsu SUNWqfsr
```

Note – The `SUNWqfsr` package must be the last package removed. The 4U1 release does not include any localized software packages.

The following example command removes the `SUNWcqfs`, the `SUNWfqfs`, and the `SUNWjqfs` localized packages from a 4U0 release:

```
# pkgrm SUNWcqfs SUNWfqfs SUNWjqfs SUNWqfs
```

Note – The `SUNWqfs` package must be the last package removed.

Adding the Upgrade Packages

The Sun StorEdge QFS software packages use the Sun Solaris packaging utilities for adding and deleting software. The `pkgadd`(1M) command prompts you to confirm various actions necessary to upgrade the Sun StorEdge QFS package.

During the installation, the system detects the presence of conflicting files and prompts you to indicate whether you want to continue with the installation. You can go to another window and copy the files that you want to save to an alternate location.

▼ To Add the Packages

1. **Use the `cd`(1) command to change to the directory where the software package release files reside.**

   This is one of the following, depending on your release media:

   - If you downloaded the release files as described in "Obtaining the Release Files" on page 19, change to the directory to which you downloaded the files.

- If you obtained the release files from a CD-ROM, change to the directory on the CD-ROM that corresponds to your OS version.

2. **Use the `pkgadd`(1M) command to upgrade the `SUNWqfsr` and `SUNWqfsu` packages.**

   For example:

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

3. **Enter `yes` or `y` in response to each of the questions.**

   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.

Installing File System Manager

Perform the task described in this section if you want to be able to use the File System Manager interface.

File System Manager is an online interface that enables you to configure many of the components in a Sun StorEdge QFS environment. You can use this tool to control, monitor, configure, and reconfigure the environment's components.

Note – File System Manager does not support file systems in Sun Cluster environments.

▼ To Install the File System Manager Software

1. **Log in to the server that you want to use as the management station.**

   This can be the same server on which you installed the `SUNWqfsr` and `SUNWqfsu` packages.

2. **Become superuser.**

3. **Use the `cd`(1) command to change to the directory where the software package release files reside on your server.**

4. **Execute the `fsmgr_setup` script to start the installation process.**

   For example:

```
# ./fsmgr_setup
```

5. **Answer the questions as prompted by the `fsmgr_setup` script.**

   During the installation procedure, you are asked questions about your environment. The script prompts you to enter passwords for the `SAMadmin` role and for the `samadmin` and `samuser` login IDs.

   The `fsmgr_setup` script automatically installs the following:

   - The Tomcat, Java Runtime Environment (JRE), JATO, and Java Web Console packages. If you have existing versions of these software packages that are not compatible with File System Manager, the installation software asks whether you want the appropriate levels to be installed at this time.
   - The `SUNWfsmgru` package.
   - The `SUNWfsmgrr` package.

   The installation scripts prompt you to specify whether you want to install localized packages.

   After installing the packages, it starts the Tomcat Web Server, enables logging, and creates the `SAMadmin` role.

6. **Log in to the Sun StorEdge QFS server and become superuser.**

7. **Use the `ps`(1) and `grep`(1) commands to make sure that the `rpcbind` service is running.**

```
# ps -ef | grep rpcbind
```

8. **Examine the output from the preceding commands.**

   The output should contain a line similar to the following:

```
root   269     1  0   Feb 08 ?        0:06 /usr/sbin/rpcbind
```

   If `rpcbind` does not appear in the output, enter the following command to start the `rcpbind` service:

```
# /usr/sbin/rpcbind
```

9. **(Optional) Start the File System Manager (`fsmgmtd`) daemon.**

   If you did not choose to start the File System Manager daemon automatically during the installation process, do one of the following:

   - Enter the following command to start the File System Manager daemon and have it restart automatically every time the daemon process dies. With this configuration, the daemon also automatically restarts at system reboot.

```
# /opt/SUNWsamfs/sbin/fsmadm config -a
```

   - Enter the following command if you want the File System Manager daemon to run only once and not automatically restart.

```
# /opt/SUNWsamfs/sbin/fsmadm start
```

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

   For information about using File System Manager see "Using the File System Manager Software" on page 28, or see the File Manager online help.

Restoring the File System

The instructions in this section cover the tasks involved in restoring the Sun StorEdge QFS file system after an upgrade.

▼ To Verify the `mcf` File

1. **Enter the `sam-fsd`(1M) command.**
2. **Examine the output for errors, as follows:**
   - If the `mcf` file is free of syntax errors, the `sam-fsd`(1M) output is similar to that shown in CODE EXAMPLE 5-7. It contains information about the file systems and other system information.

CODE EXAMPLE 5-7 `sam-fsd`(1M) Output Showing No Errors

```
# sam-fsd
Trace file controls:
sam-amld      off

sam-archiverd off

sam-catserverd off

sam-fsd       off

sam-rftd      off

sam-recycler  off

sam-sharefsd  off

sam-stagerd   off

sam-serverd   off

sam-clientd   off

sam-mgmt      off
```

- If the `mcf` file contains syntax or other errors, the errors are noted in the output.

  If your `mcf` file has errors, refer to "Setting Up the Environment Configuration" on page 30 and to the `mcf`(4) man page for information about how to create this file correctly.

Note – If you change the `mcf` file after the Sun StorEdge QFS file system is in use, you must convey the new `mcf` specifications to the Sun StorEdge QFS software. For information about propagating `mcf` file changes to the system, see the *Sun StorEdge QFS Configuration and Administration Guide*.

▼ To Modify the `/etc/vfstab` File

Perform this task if you modified the `/etc/vfstab` file in "Unmounting File Systems" on page 86.

- **Edit this file again, and change the Mount at Boot field for all Sun StorEdge QFS file systems from `no` to `yes` or `delay`.**

▼ To Reinitialize and Restore the File System

In this task, you reinitialize the file systems and restore the saved data in the new file systems. This task completes the process initiated in "Backing Up Existing File Systems" on page 83. To accomplish this, use the `sammkfs`(1M) and `qfsrestore`(1M) commands on each file system.

Caution – The Sun StorEdge QFS 4U2 and later software does not enable you to initialize a file system with a version 1 superblock. The Sun StorEdge QFS 4U2 file system allows file systems to be initialized only with the version 2 superblock. If you are upgrading from 4U0 using a version 1 superblock, be aware that issuing a 4U2 or later `sammkfs`(1M) command at this point reinitializes your file system with a version 2 superblock.

1. **Issue the `samfsinfo`(1M) command and examine the output.**

   The output tells you the DAU size that was specified with the `sammkfs`(1M) command when the file system was created. You will use this DAU size again in Step 2.

2. **Use the `sammkfs`(1M) command to initialize a new Sun StorEdge QFS file system.**

   The following example command reinitializes a file system named `qfs1` with a DAU size of 512 kilobytes:

   ```
   # sammkfs -a 512 qfs1
   ```

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

3. **Use the `qfsrestore`(1M) command to restore the dumped data in the new file system.**

   For example, suppose you had a file system named `qfs1` (mounted at `/qfs1`) that you wanted to restore from files dumped to `qfs1.bak`, which existed outside of the Sun StorEdge QFS file system. In this case, you would issue the following commands:

   ```
   # cd /qfs1
   # qfsrestore -f /save/qfs/qfs1.bak
   ```

To Check the File System

Perform this task if you did not reinitialize and restore the file system as just described.

- **Use the `samfsck`(1M) command to check each existing file system for inconsistencies.**

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

Mounting the File System

You can mount the Sun StorEdge QFS file system using File System Manager or the CLI.

▼ To Mount the File System Using File System Manager

1. **From the Servers page, click the name of the server on which the file system is located.**

   The File System Summary page is displayed.

2. **Select the radio button next to the file system that you want to mount.**

3. **From the Operations menu, choose Mount.**

▼ To Mount the File System Using the CLI

- **Issue the `mount`(1M) command.**

  In the following example, `qfs1` is the name of the file system to be mounted:

```
# mount qfs1
```

Recompiling API-Dependent Applications

File headers, the calling sequence, and other elements of the Sun StorEdge QFS application programming interface (API) can change from release to release. If you are running applications that use the API, you should recompile them all at this time.

Caution – Failure to recompile API-dependent applications at this point can cause your applications to generate unexpected results.

Upgrading the Solaris OS

The following section describes how to upgrade the Solaris OS when running the Sun StorEdge QFS software.

▼ To Upgrade the Solaris OS in a Sun StorEdge QFS Environment

Many of the steps involved in upgrading your Solaris OS level are identical to the steps involved in upgrading your Sun StorEdge QFS environment. Some of the steps in this procedure reference procedures in the previous sections.

1. **Obtain the Sun StorEdge QFS and Solaris OS software upgrades.**

   Sun StorEdge QFS software supports various levels of the Solaris OS. You should not reinstall your old Sun StorEdge QFS software on your newly upgraded Solaris OS unless you are sure they are compatible.

   Contact your application service provider or Sun Microsystems to obtain new copies of the software.

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 Sun StorEdge QFS environment. Also make sure that you have backup copies of files in the `/etc/opt/SUNWsamfs` directory.

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

   The file systems should be backed up regularly according to your site's policies and as described in "Backing Up Data" on page 54. 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. **Unmount the file systems.**

   For instructions, see "Unmounting File Systems" on page 86.

5. **Remove the existing Sun StorEdge QFS software.**

   You must remove the existing Sun StorEdge QFS package before installing either the new package or the new operating system level. For instructions, see "Removing Existing Sun StorEdge QFS Software" on page 87.

6. **Upgrade the Solaris OS.**

   Install the new Solaris OS revision using the corresponding Sun Solaris upgrade procedures.

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

   The Sun StorEdge 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 Sun StorEdge QFS package. For instructions, see "Adding the Upgrade Packages" on page 88.

8. **(Optional) Update the `mcf` file.**

   If device names have changed, it might be necessary to update the `mcf` file to match the new device names. Verify the new device names, and then follow the procedure in "Restoring the File System" on page 91.

9. If your `/etc/vfstab` file does not have `yes` in the `Mount at Boot` field, m**ount the file systems.**

   Use the procedure described in "Mounting the File System" on page 94.

APPENDIX **A**

Release Package Contents

This appendix describes the content of the release packages and shows the directories and files that the software creates when it is installed.

This appendix contains the following sections:

- "Release Package Contents" on page 97
- "Directories and Files Created" on page 98

Release Package Contents

The Sun StorEdge QFS software packages are in Sun Solaris `pkgadd`(1M) format. These packages reflect the Sun Solaris version for the platform on which you will be installing the Sun StorEdge QFS software.

TABLE A-1 shows the release packages.

TABLE A-1 Release Packages

Installed Package	Description
`SUNWqfsr, SUNWqfsu`	Sun StorEdge QFS software packages
`SUNWfsmgrr, SUNWfsmgru`	File System Manager software packages

The releases are identified using characters arranged in the following format:

major `U` *update*`.`*patch*

The "U" in this format stands for "update."

In the patch number field, a number between 1 and 99 indicates a patch release, and a letter from A through Z indicates pre-release software. The base release of a first feature release of a major release might not contain a patch level.

For example:

- 4U0 is release 4, update 0, a major release with no minor release revisions and no bug fixes.
- 4U2 is release 4, update 2, a minor release.
- 4U2.1 is a patch release that contains software fixes for a major or minor release. This number appears in the patch's README file.

Directories and Files Created

This section describes the directories and files associated with the Sun StorEdge QFS product. You can obtain additional information about the files in this section from the man pages after the software is installed.

Directories Created at Installation

TABLE A-2 lists the directories created when the Sun StorEdge QFS software packages are installed.

TABLE A-2 Directories Created

Directory	Content
`/etc/fs/samfs`	Commands specific to Sun StorEdge QFS software.
`/etc/opt/SUNWsamfs`	Configuration and license files.
`/etc/opt/SUNWsamfs/scripts`	Site-customizable scripts.
`/opt/SUNWsamfs/bin`	User command binaries.
`/opt/SUNWsamfs/client`	Files for the remote procedure call API client.
`/opt/SUNWsamfs/doc`	Documentation repository for any informational files included in the release. The README file, which summarizes the installed release's features, is included in this directory.
`/opt/SUNWsamfs/examples`	Various example configuration files.
`/opt/SUNWsamfs/include`	API include files.
`/opt/SUNWsamfs/lib`	Relocatable libraries.

TABLE A-2 Directories Created *(Continued)*

Directory	Content
/opt/SUNWsamfs/man	man(1) pages.
/opt/SUNWsamfs/mibs	Standard MIB files and product MIB (SUN-SAM-MIB.mib).
/opt/SUNWsamfs/sbin	System administrator commands and daemon binaries.
/opt/SUNWsamfs/sc	Sun Cluster binaries and configuration files.
/opt/SUNWfsmgr/bin	File System Manager administrator commands.
/opt/SUNWfsmgr/doc	File System Manager online documentation repository.
/var/opt/SUNWsamfs	Device catalogs, catalog trace file, log files, archiver data directory, and queue files.

Files Created at Installation

TABLE A-3 lists miscellaneous files created when the Sun StorEdge QFS software is installed.

TABLE A-3 Files Created - Miscellaneous

File	Description
/etc/sysevent/config/SUNW,SUNWsamfs,sysevent.conf	
	Solaris system event handler configuration file.
/kernel/drv/amd64/samaio	File system asynchronous I/O pseudo-driver (64-bit version for x64 platforms).
/kernel/drv/amd64/samioc	Sun Solaris 64-bit file system interface module (for x64 platforms).
/kernel/drv/samaio.conf	Configuration file for samaio.
/kernel/drv/samaio	File system 32-bit asynchronous I/O pseudo-driver (not present in the Solaris 10 OS).
/kernel/drv/samioc.conf	Configuration file for the samioc module.
/kernel/drv/samioc	Sun Solaris 32-bit file system interface module (not present under the Solaris 10 OS).
/kernel/drv/sparcv9/samaio	File system asynchronous I/O pseudo-driver (64-bit version for SPARC platforms).
/kernel/drv/sparcv9/samioc	Sun Solaris 64-bit file system interface module (for SPARC platforms)
/kernel/fs/amd64/samfs	Sun Solaris 64-bit file system module for x64 platforms.
/kernel/fs/samfs	Sun Solaris 32-bit file system module (not present under the Solaris 10 OS on SPARC).

TABLE A-3 Files Created - Miscellaneous *(Continued)*

File	Description
`/kernel/fs/sparcv9/samfs`	Sun Solaris 64-bit file system module for SPARC platforms.
`/opt/SUNWsamfs/sc/etc/SUNW.qfs`	Sun Cluster configuration file created only in the presence of Sun Cluster software.
`/usr/cluster/lib/rgm/rtreg/SUNW.qfs`	Sun Cluster configuration file created only in the presence of Sun Cluster software.
`/var/log/webconsole/host.conf`	File System Manager configuration file.
`/var/opt/SUNWsamfs/faults`	Faults history file.
`/var/sadm/samqfsui/fsmgr_uninstall`	Software for removing File System Manager and its supporting applications.

Note – The 32-bit modules are not distributed for the Solaris 10 OS packages on a SPARC platform.

The Sun StorEdge QFS file system has dynamically loadable components that are stored in the Sun Solaris `/kernel` directory (see TABLE A-3). You can determine the modules that are loaded by using the `modinfo`(1M) command. Typically, the kernel loads the file system module at boot time. Alternatively, you can load the file system module when the file system is first mounted after the Sun software is installed.

After the Sun StorEdge QFS software is installed, it creates files that it uses for fault notification. TABLE A-4 lists these files. When the software detects faults serious enough to merit user attention, the software uses these trap and log files to convey fault information through the File System Manager software.

TABLE A-4 Files Created - Fault Notification

File	Description
`/etc/opt/SUNWsamfs/scripts/sendtrap`	Sends trap information.
`/opt/SUNWsamfs/sbin/fault_log`	Records faults.

The software creates the files listed in TABLE A-4 with `-rwxr-x---` permissions. Do not change these file permissions. If execute permissions are lost, for example, the system writes messages such as the following to `/var/adm/messages`:

```
SUNW,SUNWsamfs,sysevent.conf, line1: no execute access to
/opt/SUNWsamfs/sbin/tapealert_trap - No such file or directory.
```

Site Files

The configuration procedures in this manual direct you to create several files. The Sun StorEdge QFS software uses these site files.

Note – Your site's configuration files must contain ASCII characters only.

There is only one site file that you are required to create at your site in order to use the Sun StorEdge QFS software. It is the master configuration (`mcf`) file `/etc/opt/SUNWsamfs/mcf`. For more information about this file, see the mcf(4) man page.

You might also create the following files, depending on the software packages you install and the features you use:

- `/etc/opt/SUNWsamfs/samfs.cmd` – File system mount parameter command file. For more information, see the `samfs.cmd`(4) man page or see the *Sun StorEdge QFS Configuration and Administration Guide*.
- `/etc/opt/SUNWsamfs/defaults.conf` – Miscellaneous default values. For more information, see the `defaults.conf`(4) man page.

Note – If you are also using the Sun StorEdge SAM-FS software for archive management, a number of additional site files are required. See the *Sun StorEdge SAM-FS Installation and Upgrade Guide* for more information.

Modified System Files

During installation, Sun StorEdge QFS software adds information to certain Sun Solaris system files. These system files are ASCII text files. The Solaris OS uses these files to identify loadable kernel modules by number rather than by name.

The Sun StorEdge QFS software adds information to the following files:

- `/etc/security/auth_attr` – This file is the authorization description database. The system adds the following lines to this file:

```
com.sun.netstorage.samqfs.web.read:::SAM-FS Read Access::
com.sun.netstorage.samqfs.web.write:::SAM-FS Write Access::
com.sun.netstorage.samqfs.web.*:::SAM-FS All Access::
```

- `/etc/user_attr` – This file is the extended user attributes database used by File System Manager. The system adds the following lines to this file:

```
SAMadmin::::type=role;auths=com.sun.netstorage.samqfs.web.*
samadmin::::type=normal;roles=SAMadmin
```

APPENDIX **B**

Uninstalling the Software

This appendix tells you how to uninstall the Sun StorEdge QFS Linux Client software and the File System Manager software. It contains the following topics:

- "Uninstalling the Sun StorEdge QFS Linux Client Software" on page 103
- "Uninstalling the File System Manager Software" on page 104

For instructions on uninstalling the Sun StorEdge QFS packages, see "Removing Existing Sun StorEdge QFS Software" on page 87.

Uninstalling the Sun StorEdge QFS Linux Client Software

- **To uninstall the Linux client software, use the uninstall script that resides in the** `/var/opt/SUNWsamfs` **directory.**

Caution – Do not use other processes, such as `rpm -e`, to uninstall the software. They can cause unexpected results and problems with uninstalling or reinstalling the software.

Uninstalling the File System Manager Software

1. **Log in to the server on which the File System Manager software is installed.**

   This is the host on which you ran the `fsmgr_setup` script at installation time.

2. **Become superuser.**

3. **Issue the following command to remove the File System Manager software and all the applications that were installed with it:**

```
# /var/sadm/samqfsui/fsmgr_uninstall
```

   This script prompts you to confirm removal of the Tomcat Web Server, JRE packages, and information pertaining to administrator and user accounts.

APPENDIX **C**

Command Reference

The Sun StorEdge QFS environment consists of a file system, daemons, processes, various types of commands (user, administrator, and so on), and tools. This appendix describes the commands that are included in the Sun StorEdge QFS software distributions.

The Sun StorEdge QFS commands operate in conjunction with the standard UNIX file system commands. All the commands are documented in UNIX `man`(1) pages.

This appendix contains the following sections:

- "User Commands" on page 105
- "General System Administrator Commands" on page 106
- "File System Commands" on page 107
- "Application Programming Interface" on page 108
- "Operational Utilities" on page 108

User Commands

By default, file system operations are transparent to the end user. Depending on your site practices, however, you might want to make some commands available to users at your site to fine-tune certain operations. TABLE C-1 summarizes these commands.

TABLE C-1 User Commands

Command	Description
`sdu`(1)	Summarizes disk usage. The `sdu`(1) command is based on the GNU version of the `du`(1) command.
`setfa`(1)	Sets file attributes.

TABLE C-1 User Commands *(Continued)*

Command	Description
sfind(1)	Searches for files in a directory hierarchy. The sfind(1) command is based on the GNU version of the find(1) command and contains options for searching based on Sun StorEdge QFS and SAM-QFS file attributes.
sls(1)	Lists contents of directories. The sls(1) command is based on the GNU version of the ls(1) command and contains options for displaying file system attributes and information.
squota(1)	Reports quota information.

General System Administrator Commands

TABLE C-2 summarizes the commands that you can use to maintain and manage the system.

TABLE C-2 General System Administrator Commands

Command	Description
fsmadm(1M)	Starts or stops the fsmgmtd daemon.
fsmgr_setup(1M)	Installs or upgrades the File System Manager software
samcmd(1M)	Executes one samu(1M) operator interface utility command.
samexplorer(1M)	Generates a Sun StorEdge QFS diagnostic report script.
samu(1M)	Invokes the full-screen, text-based operator interface. This interface is based on the curses(3CURSES) software library. The samu utility displays the status of devices and enables the operator to control automated libraries.

File System Commands

TABLE C-3 summarizes the commands that you can use to maintain the file system.

TABLE C-3 File System Commands

Commands	Description
`mount`(1M)	Mounts a file system. The man page name for this command is `mount_samfs`(1M).
`qfsdump`(1M) `qfsrestore`(1M)	Creates or restores a dump file containing the file data and metadata associated with a Sun StorEdge QFS file system.
`sambcheck`(1M)	Lists block usage for a file system.
`samchaid`(1M)	Changes the `file admin set ID` attribute. For use with quotas.
`samfsck`(1M)	Checks and repairs metadata inconsistencies in a file system and reclaims allocated, but unused, disk space.
`samfsconfig`(1M)	Displays configuration information.
`samfsdump`(1M) `samfsrestore`(1M)	Creates or restores a dump file of the metadata associated with a SAM-QFS file system.
`samfsinfo`(1M)	Displays information about the layout of a Sun StorEdge QFS or SAM-QFS file system.
`samfstyp`(1M)	Determines the Sun StorEdge QFS or SAM-QFS file system type.
`samgrowfs`(1M)	Expands a file system by adding disk devices.
`sammkfs`(1M)	Initializes a new file system from disk devices.
`samncheck`(1M)	Returns a full directory path name, given the mount point and inode number.
`samquota`(1M)	Reports, sets, or resets quota information.
`samquotastat`(1M)	Reports on active and inactive file system quotas.
`samsharefs`(1M)	Manipulates the Sun StorEdge QFS shared file system configuration information.
`samtrace`(1M)	Dumps the trace buffer.
`samunhold`(1M)	Releases SANergy file holds.
`trace_rotate`(1M)	Rotates trace files.

Application Programming Interface

You can use the application programming interface (API) to make file system requests from within a user application. The requests can be made locally or remotely to the machine on which the file system is running. The API consists of the `libsam` and `libsamrpc` libraries. These libraries contain library routines for obtaining file status; for setting archive, release, and stage attributes for a file; and for manipulating the library catalog of an automated library. The `sam-rpcd` remote procedure call daemon handles remote requests. To automatically start the `sam-rpcd` daemon, set `samrpc=on` in the `defaults.conf` file.

For more information about the API, see the `intro_libsam`(3) man page. This man page provides overview information for using the library routines in `libsam` and `libsamrpc`.

Operational Utilities

Within the Sun StorEdge QFS environment, you can use the `samu`(1M) operator utility and File System Manager to perform basic operations. TABLE C-4 summarizes the operational tools.

TABLE C-4 Operational Utilities

GUI Tools	Description
File System Manager	Provides a web-based graphical user interface to the Sun StorEdge QFS software. You can use this interface to configure, control, monitor, and reconfigure the components of your Sun StorEdge QFS environment. For information on installing File System Manager, see "Installing and Using the File System Manager Software" on page 26. For information on using the File System Manager, see its online help.
`samu`(1M)	Provides the starting point for accessing the `samu`(1M) operator utility.

APPENDIX **D**

mcf File Examples

The master configuration file, /etc/opt/SUNWsamfs/mcf, defines the topology of the equipment managed by the Sun StorEdge QFS file system. This file specifies the devices and file systems included in the environment and contains information that enables you to identify the disk slices to be used and to organize them into Sun StorEdge QFS file systems.

This appendix provides some specific examples of mcf files for various types of file systems. It contains the following sections:

- "Configuration Examples for Local File Systems" on page 109
- "Configuration Example for a Shared File System on a Solaris OS Platform" on page 118
- "Configuration Examples for Highly Available File Systems" on page 122
- "Configuration Example for a Shared File System on a Sun Cluster Platform" on page 124

Configuration Examples for Local File Systems

Use the configuration examples in this section for configuring the mcf file for a Sun StorEdge QFS file system to be installed on a single Solaris OS host.

For mcf examples that you can use in a Sun Cluster environment, see "Configuration Examples for Highly Available File Systems" on page 122.

Configuration Example 1

This example shows how to configure two Sun StorEdge QFS file systems using a server that has a Sun StorEdge Multipack desktop array connected by a SCSI attachment.

You can use the `format`(1M) command to determine how the disks are partitioned. CODE EXAMPLE D-1 shows the `format`(1M) command's output.

Note – Only the last lines of `format`(1M) output are shown.

CODE EXAMPLE D-1 `format`(1M) Command Output for Configuration Example 1

```
# format < /dev/null
Searching for disks...done

AVAILABLE DISK SELECTIONS:
       0. c0t10d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /sbus@3,0/SUNW,fas@3,8800000/sd@a,0
       1. c0t11d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /sbus@3,0/SUNW,fas@3,8800000/sd@b,0
       2. c6t2d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
          /pci@7,4000/SUNW,isptwo@3/sd@2,0
       3. c6t3d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
          /pci@7,4000/SUNW,isptwo@3/sd@3,0
       4. c6t4d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
          /pci@7,4000/SUNW,isptwo@3/sd@4,0
       5. c6t5d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
          /pci@7,4000/SUNW,isptwo@3/sd@5,0
       6. c8t2d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
          /pci@b,4000/SUNW,isptwo@3/sd@2,0
       7. c8t3d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
          /pci@b,4000/SUNW,isptwo@3/sd@3,0
       8. c8t4d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
          /pci@b,4000/SUNW,isptwo@3/sd@4,0
       9. c8t5d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
          /pci@b,4000/SUNW,isptwo@3/sd@5,0
Specify disk (enter its number):

# format /dev/rdsk/c6t2d0s2
.
.
.
Part      Tag    Flag     Cylinders        Size            Blocks
  0 unassigned    wm       0               0         (0/0/0)           0
  1 unassigned    wm       0               0         (0/0/0)           0
  2     backup    wu       0 - 4923        8.43GB    (4924/0/0) 17682084
```

CODE EXAMPLE D-1 `format`(1M) Command Output for Configuration Example 1 *(Continued)*

```
  3 unassigned    wm       0                0         (0/0/0)           0
  4 unassigned    wm       0 - 1229         2.11GB    (1230/0/0)  4416930
  5 unassigned    wm    1230 - 2459         2.11GB    (1230/0/0)  4416930
  6 unassigned    wm    2460 - 3689         2.11GB    (1230/0/0)  4416930
  7 unassigned    wm    3690 - 4919         2.11GB    (1230/0/0)  4416930
```

▼ To Configure the System

Begin writing the `mcf` file for this configuration example by defining the file system and its disk partitions, as follows:

1. **Write the `mcf` file.**

   a. **Make an `ma` entry for the first file system (`qfs1`).**

   b. **Using the information from the output of the `format` command, make an `mm` entry listing the partitions that constitute the metadata for the `qfs1` file system.**

   c. **Using the information from the output of the `format` command, make a series of `mr` entries listing the partitions that constitute the file data for the `qfs1` file system.**

   d. **Make similar entries for the second file system (`qfs2`).**

   The finished `mcf` file defines the following two file systems:

   - The `qfs1` file system, which is created on slice `4` of the following disks: `c8t2d0` (metadata), `c6t2d0` (file data), and `c6t3d0` (file data).
   - The `qfs2` file system, which is created on slice `5` of the following disks: `c8t2d0` (metadata), `c6t2d0` (file data), and `c6t3d0` (file data).

   CODE EXAMPLE D-2 shows the resulting `mcf` file.

CODE EXAMPLE D-2 `mcf` File for Sun StorEdge QFS Example 1

```
# cat /etc/opt/SUNWsamfs/mcf
#
# Equipment          Eq   Eq      Family   Device   Additional
# Identifier         Ord  Type     Set     State    Parameters
#-----------         ---  ----   ------   ------   ----------
#
qfs1                  10    ma    qfs1       on
/dev/dsk/c8t2d0s4     11    mm    qfs1       on
/dev/dsk/c6t2d0s4     12    mr    qfs1       on
/dev/dsk/c6t3d0s4     13    mr    qfs1       on
#
qfs2                  20    ma    qfs2       on
```

CODE EXAMPLE D-2 `mcf` File for Sun StorEdge QFS Example 1 *(Continued)*

```
/dev/dsk/c8t2d0s5     21     mm    qfs2         on
/dev/dsk/c6t2d0s5     22     mr    qfs2         on
/dev/dsk/c6t3d0s5     23     mr    qfs2         on
```

2. **Modify the** `/etc/vfstab` **file.**

   Make entries in the `/etc/vfstab` file for the `qfs1` and `qfs2` file systems that you defined in the `mcf` file. The last two lines in CODE EXAMPLE D-3 show entries for these new file systems.

   For a description of the fields in the `/etc/vfstab` file, see "Fields in the `/etc/vfstab` File" on page 35.

CODE EXAMPLE D-3 `/etc/vfstab` File for Sun StorEdge QFS Example 1

```
# cat /etc/vfstab
# device            device                        file            mount
# to                to                  mount     system   fsck   at     mount
# mount             fsck                point     type     pass   boot   params
# -----             ----                -----     ----     ----   ----   ------
fd                  -                   /dev/fd   fd       -      no     -
/proc               -                   /proc     proc     -      no     -
/dev/dsk/c0t10d0s1  -                   -         swap     -      no     -
/dev/dsk/c0t10d0s0  /dev/rdsk/c0t10d0s0 /         ufs      1      no     logging
swap                -                   /tmp      tmpfs    -      yes    -
qfs1                -                   /qfs1     samfs    -      yes    stripe=1
qfs2                -                   /qfs2     samfs    -      yes    stripe=1
```

Configuration Example 2

This example illustrates the configuration of a Sun StorEdge QFS file system that uses round-robin allocation on four disk drives.

This example assumes the following:

- The metadata device is a single partition (`s1`) used on controller 8, disk 4.
- The data devices consist of four disks attached to controller 6. Each disk is on a separate target (1–4).

▼ To Configure the System

This example introduces the round-robin data layout. For more information about data layout, see the *Sun StorEdge QFS Configuration and Administration Guide*.

1. **Write the `mcf` file as described in "Configuration Example 1" on page 110.**

   CODE EXAMPLE D-4 shows the `mcf` file for this round-robin disk configuration.

CODE EXAMPLE D-4 `mcf` File for Sun StorEdge QFS Example 2

```
# cat /etc/opt/SUNWsamfs/mcf
#
# Equipment          Eq   Eq     Family   Device   Additional
# Identifier         Ord  Type    Set     State    Parameters
#------------        ---  ----   ------   ------   ----------
#
qfs3                  10    ma    qfs3       on
/dev/dsk/c8t4d0s4     11    mm    qfs3       on
/dev/dsk/c6t2d0s4     12    mr    qfs3       on
/dev/dsk/c6t3d0s4     13    mr    qfs3       on
/dev/dsk/c6t4d0s4     14    mr    qfs3       on
/dev/dsk/c6t5d0s4     15    mr    qfs3       on
```

2. **Modify the `/etc/vfstab` file.**

   Edit the `/etc/vfstab` file to explicitly set round-robin allocation on the file system by specifying `stripe=0` in the `mount params` field. CODE EXAMPLE D-5 shows `stripe=0` for the `qfs3` file system.

   For a description of the fields in the `/etc/vfstab` file, see "Fields in the /etc/vfstab File" on page 35.

CODE EXAMPLE D-5 `/etc/vfstab` File for Sun StorEdge QFS Example 2

```
# cat /etc/vfstab
#device             device                          file          mount
#to                 to                   mount      system  fsck  at    mount
#mount              fsck                 point      type    pass  boot  params
#-----              ----                 -----      ----    ----  ----  ------
fd                  -                    /dev/fd    fd      -     no    -
/proc               -                    /proc      proc    -     no    -
/dev/dsk/c0t10d0s1  -                    -          swap    -     no    -
/dev/dsk/c0t10d0s0  /dev/rdsk/c0t10d0s0  /          ufs     1     no    logging
swap                -                    /tmp       tmpfs   -     yes   -
qfs3                -                    /qfs3      samfs   -     yes   stripe=0
```

3. **Initialize the Sun StorEdge QFS file system by using the `sammkfs`(1M) command.**

   The default disk allocation unit (DAU) is 64 kilobytes, but the following example sets the DAU size to 128 kilobytes:

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

Configuration Example 3

This example illustrates the configuration of a Sun StorEdge QFS file system that stripes file data to four disk drives. This example assumes the following:

- The metadata device is a single partition (`s6`) used on controller 0, LUN 0.
- The data devices consist of four disks attached to controller 6. Each disk is on a separate target (2–5).

▼ To Configure the System

1. **Write the `mcf` file as shown in "Configuration Example 1" on page 110.**

   CODE EXAMPLE D-6 shows the `mcf` file for this striped disk configuration.

CODE EXAMPLE D-6 `mcf` File for Sun StorEdge QFS Example 3

```
# Equipment          Eq   Eq     Family   Device   Additional
# Identifier         Ord  Type    Set      State    Parameters
#------------        ---  ----   ------   ------   ----------
#
qfs4                  40    ma    qfs4       on
/dev/dsk/c8t4d0s4     41    mm    qfs4       on
/dev/dsk/c6t2d0s4     42    mr    qfs4       on
/dev/dsk/c6t3d0s4     43    mr    qfs4       on
/dev/dsk/c6t4d0s4     44    mr    qfs4       on
/dev/dsk/c6t5d0s4     45    mr    qfs4       on
```

2. **Modify the `/etc/vfstab` file.**

   Set the stripe width by using the `stripe=` option. CODE EXAMPLE D-7 shows the `/etc/vfstab` file with a mount parameter of `stripe=1` set for the `qfs4` file system.

   For a description of the fields in the `/etc/vfstab` file, see "Fields in the `/etc/vfstab` File" on page 35.

CODE EXAMPLE D-7 `/etc/vfstab` File for Sun StorEdge QFS Example 3

```
# cat /etc/vfstab
#
#device             device                          file         mount
#to                 to                   mount      system fsck  at     mount
#mount              fsck                 point      type   pass  boot   params
#-----              ----                 -----      -----  ----  ----   ------
fd                  -                    /dev/fd    fd     -     no     -
/proc               -                    /proc      proc   -     no     -
/dev/dsk/c0t10d0s1  -                    -          swap   -     no     -
```

CODE EXAMPLE D-7 `/etc/vfstab` File for Sun StorEdge QFS Example 3 *(Continued)*

```
/dev/dsk/c0t10d0s0   /dev/rdsk/c0t10d0s0   /        ufs     1     no     logging
swap                 -                     /tmp     tmpfs   -     yes    -
qfs4                 -                     /qfs4    samfs   -     yes    stripe=1
```

The `stripe=1` specification stripes file data across all four of the `mr` data disks with a stripe width of one DAU. The DAU is the allocation unit you set when you use the `sammkfs`(1M) command to initialize the file system.

3. **Initialize the Sun StorEdge QFS file system by using the `sammkfs`(1M) command.**

   The following example sets the DAU size to 128 kilobytes:

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

With this striped disk configuration, any file written to this file system is striped across all of the devices in increments of 128 kilobytes. Files less than the aggregate stripe width times the number of devices still use 128 kilobytes of disk space. Files larger than 128 kilobytes have space allocated for them as needed in total space increments of 128 kilobytes.

Configuration Example 4

Striped groups enable you to build RAID-0 devices of separate disk devices. With striped groups, however, there is only one DAU per striped group. This method of writing large, effective DAUs across RAID devices saves system update time and supports high-speed sequential I/O. Striped groups are useful for writing very large files to groups of disk devices.

Note – A DAU is the minimum disk space allocated. The minimum disk space allocated in a striped group is as follows:

allocation-unit x *number of disks in the group*

Writing a single byte of data consumes a DAU on every member of the striped group. Make sure that you understand the effects of using striped groups with your file system.

The devices within a striped group must be the same size. It is not possible to increase the size of a striped group. You can add additional striped groups to the file system, however.

This example illustrates the configuration of a Sun StorEdge QFS file system that separates the metadata onto a low-latency disk. The `mcf` file defines two striped groups on four drives. This example assumes the following:

- The metadata device is a single partition (`s5`) used on controller 8, disk 4.
- The data devices consist of four disks (two groups of two identical disks) attached to controller 6. Each disk is on a separate target (2–5).

▼ To Configure the System

1. **Write the `mcf` file as shown in "Configuration Example 1" on page 110.**

   CODE EXAMPLE D-8 shows a sample `mcf` file for a striped group configuration.

CODE EXAMPLE D-8 `mcf` File for Sun StorEdge QFS Example 4

```
# cat /etc/opt/SUNWsamfs/mcf
#
# Equipment          Eq   Eq     Family   Device   Additional
# Identifier         Ord  Type    Set     State    Parameters
#-----------         ---  ----   ------   ------   ----------
#
qfs5                  50    ma    qfs5       on
/dev/dsk/c8t4d0s5     51    mm    qfs5       on
/dev/dsk/c6t2d0s5     52    g0    qfs5       on
/dev/dsk/c6t3d0s5     53    g0    qfs5       on
/dev/dsk/c6t4d0s5     54    g1    qfs5       on
/dev/dsk/c6t5d0s5     55    g1    qfs5       on
```

2. **Modify the `/etc/vfstab` file.**

   Set the stripe width by using the `stripe=` option. CODE EXAMPLE D-9 shows the `/etc/vfstab` file with a mount parameter of `stripe=0`, which specifies round-robin allocation between striped group `g0` and striped group `g1`.

   For a description of the fields in the `/etc/vfstab` file, see "Fields in the /etc/vfstab File" on page 35.

CODE EXAMPLE D-9 `/etc/vfstab` File for Sun StorEdge QFS Example 4

```
# cat /etc/vfstab
#device             device                          file          mount
#to                 to                    mount     system  fsck  at    mount
#mount              fsck                  point     type    pass  boot  params
#-----              ----                  -----     ----    ----  ----  ------
fd                  -                     /dev/fd   fd      -     no    -
/proc               -                     /proc     proc    -     no    -
/dev/dsk/c0t10d0s1  -                     -         swap    -     no    -
```

CODE EXAMPLE D-9 `/etc/vfstab` File for Sun StorEdge QFS Example 4 *(Continued)*

```
/dev/dsk/c0t10d0s0   /dev/rdsk/c0t10d0s0   /        ufs      1     no     logging
swap                 -                     /tmp     tmpfs    -     yes    -
qfs5                 -                     /qfs5    samfs    -     yes    stripe=0
```

3. **Initialize the Sun StorEdge QFS file system by using the `sammkfs`(1M) command.**

   The `-a` option is not used with striped groups because the DAU is equal to the size of an allocation, or the size, of each group.

```
# sammkfs qfs5
```

   In this example, there are two striped groups, `g0` and `g1`. With `stripe=0` in `/etc/vfstab`, devices 12 and 13 are striped; devices 14 and 15 are striped; and files are round-robined around the two striped groups. A striped group is treated as a bound entity. After you configure a stripe group, you cannot change it without issuing another `sammkfs`(1M) command.

Configuration Example for a Shared File System on a Solaris OS Platform

FIGURE D-1 illustrates a Sun StorEdge QFS shared file system configuration in a SAM-QFS environment.

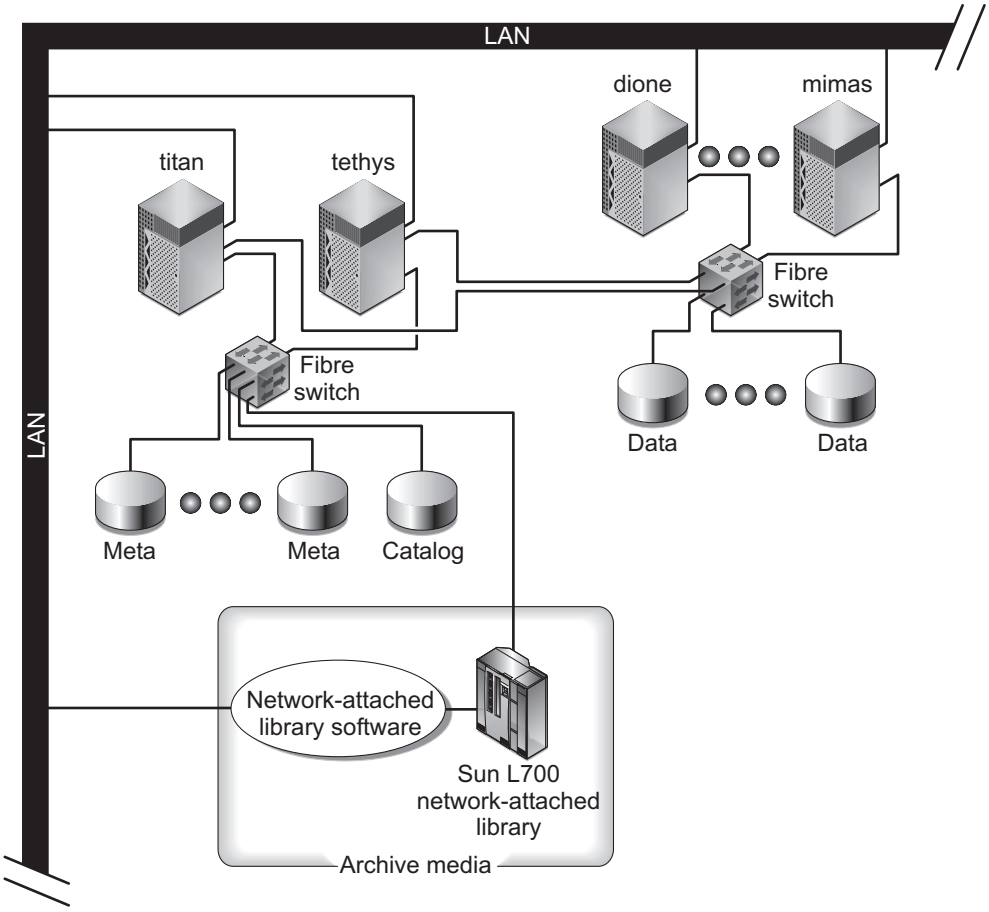

FIGURE D-1 Sun StorEdge QFS Shared File System Configuration in a SAM-QFS Environment

FIGURE D-1 shows four network-attached hosts: `titan`, `tethys`, `dione`, and `mimas`. The `tethys`, `dione`, and `mimas` hosts are clients, and `titan` is the current metadata server. The `tethys` host is a potential metadata server.

The archive media consist of a network-attached library and tape drives that are fibre-attached to `titan` and `tethys`. In addition, the archive media catalog resides in a file system that is mounted on the current metadata server, `titan`.

Metadata travels to and from the clients to the metadata server over the network. The metadata server makes all modifications to the name space, thereby keeping the metadata consistent. The metadata server also provides the locking capability, the block allocation, and the block deallocation.

Several metadata disks are connected to `titan` and `tethys` and can be accessed only by the potential metadata servers. If `titan` were unavailable, you could change the metadata server to `tethys`, and the library, tape drives, and catalog could be accessed by `tethys` as part of the Sun StorEdge QFS shared file system. The data disks are connected to all four hosts by a Fibre Channel (FC) connection.

▼ To Configure the System

1. **Issue the `format`(1M) command and examine its output.**

   Make sure that the metadata disk partitions configured for the Sun StorEdge QFS shared file system mount point are connected to the potential metadata servers. Also make sure that the data disk partitions configured for the Sun StorEdge QFS shared file system are connected to the potential metadata servers and to all the client hosts in this file system.

   If your host supports multipath I/O drivers, individual devices shown in the output of the `format`(1M) command might display multiple controllers. These correspond to the multiple paths to the actual devices.

   CODE EXAMPLE D-10 shows the `format`(1M) command output for `titan`. There is one metadata disk on controller 2, and there are three data disks on controller 3.

CODE EXAMPLE D-10 `format` (1M) Command Output on `titan`

```
titan<28>format
Searching for disks...done


AVAILABLE DISK SELECTIONS:
       0. c1t0d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w2100002037e9c296,0
       1. c2t2100002037E2C5DAd0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w2100002037e9c296,0
       2. c2t50020F23000065EEd0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w50020f23000065ee,0
       3. c3t50020F2300005D22d0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@8,600000/SUNW,qlc@1/fp@0,0/ssd@w50020f2300005d22,0
       4. c3t50020F2300006099d0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
```

CODE EXAMPLE D-10 `format` (1M) Command Output on `titan` *(Continued)*

```
          /pci@8,600000/SUNW,qlc@1/fp@0,0/ssd@w50020f2300006099,0
       5. c3t50020F230000651Cd0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@8,600000/SUNW,qlc@1/fp@0,0/ssd@w50020f230000651c,0
```

CODE EXAMPLE D-11 shows the `format`(1M) command output for `tethys`. There is one metadata disk on controller 2, and there are four data disks on controller 7.

CODE EXAMPLE D-11 `format` (1M) Command Output on `tethys`

```
tethys<1>format
Searching for disks...done


AVAILABLE DISK SELECTIONS:
       0. c0t1d0 <IBM-DNES-318350Y-SA60 cyl 11112 alt 2 hd 10 sec 320>
          /pci@1f,4000/scsi@3/sd@1,0
       1. c2t2100002037E9C296d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w2100002037e9c296,0
       2. c2t50020F23000065EEd0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@1f,4000/SUNW,qlc@4/ssd@w50020f23000065ee,0
       3. c7t50020F2300005D22d0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@1f,4000/SUNW,qlc@5/ssd@w50020f2300005d22,0
       4. c7t50020F2300006099d0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@1f,4000/SUNW,qlc@5/ssd@w50020f2300006099,0
       5. c7t50020F230000651Cd0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@1f,4000/SUNW,qlc@5/ssd@w50020f230000651c,0
```

Note the following in CODE EXAMPLE D-11:

- The data disks on `titan`'s controller 3 are the same disks as on `tethys`'s controller 7. You can verify this by looking at the World Wide Name, which is the last component in the device name. For `titan`'s number 3 disk, the World Wide Name is `50020f2300005d22`. This is the same name as number 3 on controller 7 on `tethys`.
- For `titan`'s metadata disk, the World Wide Name is `50020F23000065EE`. This is the same metadata disk as `tethys`'s controller 2, target 0.

CODE EXAMPLE D-12 shows the `format`(1M) command output for `mimas`. This shows three data disks on controller 1 and no metadata disks.

CODE EXAMPLE D-12 `format` (1M) Command Output on `mimas`

```
mimas<9>format
Searching for disks...done


AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
          /pci@1f,4000/scsi@3/sd@0,0
       1. c1t50020F2300005D22d0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@1f,4000/SUNW,qlc@4/fp@0,0/ssd@w50020f2300005d22,0
       2. c1t50020F2300006099d0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@1f,4000/SUNW,qlc@4/fp@0,0/ssd@w50020f2300006099,0
       3. c1t50020F230000651Cd0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@1f,4000/SUNW,qlc@4/fp@0,0/ssd@w50020f230000651c,0
```

As shown in CODE EXAMPLE D-11 and CODE EXAMPLE D-12, the data disks on `titan`'s controller 3 are the same disks as those on `mimas`'s controller 1. You can verify this by looking at the World Wide Name, which is the last component in the device name.

Note – All the data disk partitions must be connected and accessible from all the hosts that share this file system. All the disk partitions, for both data and metadata, must be connected and accessible to all potential metadata servers. You can use the `format`(1M) command to verify these connections.

For some storage devices, it is possible that the `format`(1M) command's output does not present unique worldwide Names. If you find that this is the case, see the `libdevid`(3LIB) man page for information about finding such devices on different hosts.

---

2. **Use `vi`(1) or another editor to create the `mcf` file on the metadata server.**

   The only difference between the `mcf` file of a shared Sun StorEdge QFS file system and an unshared Sun StorEdge QFS file system is the presence of the `shared` keyword in the Additional Parameters field of the file system name line for a Sun StorEdge QFS shared file system.

Note – If Sun StorEdge QFS or SAM-QFS file systems are already operational on the Sun StorEdge QFS shared file system's metadata server or on any of the client host systems, select a Family Set name and select equipment ordinals that do not conflict with existing Family Set names or equipment ordinals on any host that will be included in the Sun StorEdge QFS shared file system.

CODE EXAMPLE D-13 shows a fragment of the `mcf` file for `titan` that defines several disks for use in the Sun StorEdge QFS shared file system. It shows the `shared` keyword in the Additional Parameters field on the file system name line.

CODE EXAMPLE D-13 Sun StorEdge QFS Shared File System `mcf` File Example for `titan`

```
# Equipment                       Eq   Eq    Family   Dev   Addl
# Identifier                      Ord  Type  Set      Stat  Params
------------                      ---  ----  ------   ----  ------
sharefs1                          10   ma    sharefs1 on    shared
/dev/dsk/c2t50020F23000065EEd0s6 11   mm    sharefs1 on
/dev/dsk/c3t50020F2300005D22d0s6 12   mr    sharefs1 on
/dev/dsk/c3t50020F2300006099d0s6 13   mr    sharefs1 on
/dev/dsk/c3t50020F230000651Cd0s6 14   mr    sharefs1 on
```

Configuration Examples for Highly Available File Systems

The Sun Cluster software moves a Sun StorEdge QFS highly available file system from a failing node to a viable node in the event of a node failure.

Each node in the Sun Cluster environment that can host this file system must have an `mcf` file. During the file system configuration process, you copy `mcf` file lines from the metadata server's `mcf` file to other nodes in the Sun Cluster environment. For more information, see "Editing `mcf` Files on Other Hosts" on page 59.

▼ To Create an `mcf` File for a Highly Available File System

The procedure for creating an `mcf` file for a Sun StorEdge QFS highly available file system is as follows:

1. **Make an `ma` entry for the file system.**

2. **Make an `mm` entry listing the partitions that constitute the metadata for the `qfs1` file system.**

3. **Make a series of `mr`, `gXXX`, or `md` entries listing the partitions that constitute the file data for the `qfs1` file system.**

   You can use the `scdidadm`(1M) command to determine the partitions to use.

Example 1. CODE EXAMPLE D-14 shows an `mcf` file entry for a Sun StorEdge QFS highly available file system that uses raw devices.

CODE EXAMPLE D-14 `mcf` File That Specifies Raw Devices

```
Equipment             Eq   Eq     Family   Additional
Identifier            Ord  Type   Set      Parameters
--------------------- ---  ----   ------   ----------
qfs1                    1   ma    qfs1     on
/dev/global/dsk/d4s0   11   mm    qfs1
/dev/global/dsk/d5s0   12   mr    qfs1
/dev/global/dsk/d6s0   13   mr    qfs1
/dev/global/dsk/d7s0   14   mr    qfs1
```

Example 2. CODE EXAMPLE D-15 shows an `mcf` file entry for a Sun StorEdge QFS highly available file system that uses Solaris Volume Manager metadevices. In this example, the Solaris Volume Manager metaset in use is named `red`.

CODE EXAMPLE D-15 `mcf` File That Specifies Solaris Volume Manager Devices

```
Equipment             Eq   Eq     Family   Additional
Identifier            Ord  Type   Set      Parameters
--------------------- ---  ----   ------   ----------
qfs1                    1   ma    qfs1     on
/dev/md/red/dsk/d0s0   11   mm    qfs1
/dev/md/red/dsk/d1s0   12   mr    qfs1
```

Example 3. CODE EXAMPLE D-16 shows an `mcf` file entry for a Sun StorEdge QFS highly available file system that uses VxVm devices.

CODE EXAMPLE D-16 `mcf` File That Specifies VxVM Devices

```
Equipment             Eq   Eq     Family   Additional
Identifier            Ord  Type   Set      Parameters
--------------------- ---  ----   ------   ----------
qfs1                    1   ma    qfs1     on
/dev/vx/dsk/oradg/m1   11   mm    qfs1
/dev/vx/dsk/oradg/m2   12   mr    qfs1
```

Configuration Example for a Shared File System on a Sun Cluster Platform

In this example, `ash` and `elm` are nodes in a Sun Cluster environment. Host `ash` is the metadata server. The keyword `shared` in this example's `mcf` file indicates to the system that this is a shared file system. This example builds upon "Example: Verifying Devices and Device Redundancy" on page 12.

▼ To Create an `mcf` File for a Shared File System in a Sun Cluster Environment

You must create the `mcf` file on the node that you want to designate as the metadata server. The procedure for creating an `mcf` file for a Sun StorEdge QFS shared file system in a Sun Cluster environment is as follows:

1. **Use the `scdidadm`(1M) `-L` command to obtain information about the devices included in the Sun Cluster environment.**

   The `scdidadm`(1M) command administers the device identifier (DID) devices. The `-L` option lists all the DID device paths, including those on all nodes in the Sun Cluster environment.

   CODE EXAMPLE D-17 uses Sun StorEdge T3 arrays in a RAID-5 configuration. The output shows that you can use devices 4 through 9 for configuring the disk cache for a shared file system.

CODE EXAMPLE D-17 `scdidadm`(1M) Command Example

```
ash# scdidadm -L
1        ash:/dev/rdsk/c0t6d0        /dev/did/rdsk/d1
2        ash:/dev/rdsk/c1t1d0        /dev/did/rdsk/d2
3        ash:/dev/rdsk/c1t0d0        /dev/did/rdsk/d3
4        elm:/dev/rdsk/c6t50020F2300004921d1 /dev/did/rdsk/d4
4        ash:/dev/rdsk/c5t50020F2300004921d1 /dev/did/rdsk/d4
5        elm:/dev/rdsk/c6t50020F2300004921d0 /dev/did/rdsk/d5
5        ash:/dev/rdsk/c5t50020F2300004921d0 /dev/did/rdsk/d5
6        elm:/dev/rdsk/c6t50020F23000049CBd1 /dev/did/rdsk/d6
6        ash:/dev/rdsk/c5t50020F23000049CBd1 /dev/did/rdsk/d6
7        elm:/dev/rdsk/c6t50020F23000049CBd0 /dev/did/rdsk/d7
7        ash:/dev/rdsk/c5t50020F23000049CBd0 /dev/did/rdsk/d7
8        elm:/dev/rdsk/c6t50020F23000055A8d0 /dev/did/rdsk/d8
8        ash:/dev/rdsk/c5t50020F23000055A8d0 /dev/did/rdsk/d8
9        elm:/dev/rdsk/c6t50020F23000078F1d0 /dev/did/rdsk/d9
```

CODE EXAMPLE D-17 `scdidadm`(1M) Command Example *(Continued)*

```
9         ash:/dev/rdsk/c5t50020F23000078F1d0 /dev/did/rdsk/d9
10        elm:/dev/rdsk/c0t6d0        /dev/did/rdsk/d10
11        elm:/dev/rdsk/c1t1d0        /dev/did/rdsk/d11
12        elm:/dev/rdsk/c1t0d0        /dev/did/rdsk/d12
```

2. **Using the output from the `scdidadm`(1M) `-L` command, use the `format`(1M) command to display the information for the devices in the Sun Cluster environment.** CODE EXAMPLE D-18 **shows the `format` command output from all the `/dev/did` devices. You will need this information when you build the `mcf` file.**

CODE EXAMPLE D-18 `format`(1M) Command Output

```
ash# format /dev/did/rdsk/d4s2
selecting /dev/did/rdsk/d4s2

Primary label contents:

Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 64 sec 32>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =   64
nsect       =   32
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264       16.86GB    (17265/0/0) 35358720
  1        usr    wm   17265 - 34529       16.86GB    (17265/0/0) 35358720
  2     backup    wu       0 - 34529       33.72GB    (34530/0/0) 70717440
  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

ash# format /dev/did/rdsk/d5s2
selecting /dev/did/rdsk/d5s2

Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 192 sec 64>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =  192
nsect       =   64
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264      101.16GB    (17265/0/0) 212152320
  1        usr    wm   17265 - 34529      101.16GB    (17265/0/0) 212152320
```

CODE EXAMPLE D-18 `format`(1M) Command Output *(Continued)*

```
  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

ash# format /dev/did/rdsk/d6s2
selecting /dev/did/rdsk/d6s2

Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 64 sec 32>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =   64
nsect       =   32
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264       16.86GB    (17265/0/0) 35358720
  1        usr    wm   17265 - 34529       16.86GB    (17265/0/0) 35358720
  2     backup    wu       0 - 34529       33.72GB    (34530/0/0) 70717440
  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


ash# format /dev/did/rdsk/d7s2
selecting /dev/did/rdsk/d7s2

Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 192 sec 64>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =  192
nsect       =   64
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264      101.16GB    (17265/0/0) 212152320
  1        usr    wm   17265 - 34529      101.16GB    (17265/0/0) 212152320
  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
```

CODE EXAMPLE D-18 `format`(1M) Command Output *(Continued)*

```
ash# format /dev/did/rdsk/d8s2
selecting /dev/did/rdsk/d8s2

Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 128 sec 128>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =  128
nsect       =  128
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264      134.88GB    (17265/0/0) 282869760
  1        usr    wm   17265 - 34529      134.88GB    (17265/0/0) 282869760
  2     backup    wm       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


ash# format /dev/did/rdsk/d9s2
selecting /dev/did/rdsk/d9s2

Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 128 sec 128>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =  128
nsect       =  128
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264      134.88GB    (17265/0/0) 282869760
  1        usr    wm   17265 - 34529      134.88GB    (17265/0/0) 282869760
  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
```

The `format`(1M) command reveals the space available on a device, but it does not reveal whether a disk is mirrored or striped. CODE EXAMPLE D-18's `format`(1M) output reveals the following information that is used during creation of the `mcf` file shown in CODE EXAMPLE D-19:

- Output for devices `d4s0` and `d6s0` shows 16.86 Gbytes each. These devices are assigned equipment ordinal `501` and Equipment Ordinal `502`, respectively, in the `mcf` file. They are the appropriate size to use for metadata slices.
- Output for devices `d8s0` and `d9s0` shows 134.88 Gbytes each. These devices are assigned equipment ordinal `503` and Equipment Ordinal `504`, respectively, in the `mcf` file. They are the are the appropriate size to be used for storing data.

3. **Make an `ma` entry for the file system.**

   In this line entry, include the `shared` keyword in the Additional Parameters field.

4. **Make an `mm` entry listing the partitions that constitute the metadata for the `qfs1` file system.**

   Put the file system's `mm` devices on mirrored (RAID-1) disks. The `mm` devices should constitute about 10% of the space allocated for the entire file system.

5. **Make a series of `mr` entries listing the partitions that constitute the file data for the `qfs1` file system.**

   CODE EXAMPLE D-19 shows the resulting `mcf` file.

CODE EXAMPLE D-19 `mcf` File on Metadata Server `ash`

```
Equipment             Eq   Eq     Family   Additional
Identifier            Ord  Type   Set      Parameters
--------------------- ---  ----   ------   ----------
#
# Family Set sqfs1 (shared FS for SunCluster)
#
sqfs1                 500   ma    sqfs1    shared
/dev/did/dsk/d4s0     501   mm    sqfs1    -
/dev/did/dsk/d6s0     502   mm    sqfs1    -
/dev/did/dsk/d8s0     503   mr    sqfs1    -
/dev/did/dsk/d9s0     504   mr    sqfs1    -
```

Glossary

A

addressable storage The storage space encompassing online, nearline, offsite, and offline storage that is user-referenced through a Sun StorEdge QFS or Sun StorEdge SAM-FS 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.

archiver The archive program that automatically controls the copying of files to removable cartridges.

archive storage Copies of file data that have been created on archive media.

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, such as 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 In a 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*.

device logging A configurable feature that provides device-specific error information used to analyze device problems.

device scanner Software 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 Sun StorEdge SAM-FS software.

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 In a Sun SAM-Remote configuration, the buffer on the server system that is used for archiving data from the client to the server.

disk cache The disk-resident portion of the file system software, 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 threshold The maximum or minimum level of disk cache utilization, as defined by an administrator. 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 *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/second LAN.

extent array The array within a file's inode that defines the disk location of each data block assigned to the file.

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. See also *storage family set*.

FDDI Fiber-distributed data interface is a standard for data transmission in a local area network that can extend in range up to 200 km (124 miles). The FDDI protocol is based on the token ring protocol.

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.

file system A hierarchical collection of files and directories.

file-system-specific directives Archiver and releaser directives that follow global directives in the `archiver.cmd` file, are specific to a particular file system, and begin with `fs =`. File-system-specific directives apply until the next `fs =` directive line or 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, the amount of time for which a user is allowed to create files and allocate storage after reaching the soft limit.

H

hard limit For disk quotas, the 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. 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. Inodes are 512 bytes long. The inode file is a metadata file, which is separated from file data in the file system.

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 A function that grants a client host permission to perform an operation on a file for a specified period of time. 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 system 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) in a file system environment.

media Tape or optical disk cartridges.

media recycling The process of recycling or reusing archive media with few active files.

metadata Data about data. Metadata is the index information used 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 device A device (for example, a solid-state disk or mirrored device) upon which file system metadata is stored. Having file data and metadata on separate devices can increase performance. In the `mcf`(4) 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 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 with the `mount`(1M) command. The single-writer host is specified with the `-o writer` option with 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 takes somewhat longer to access.

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 Sun StorEdge SAM-FS file system interfaces with the vendor software using a Sun StorEdge SAM-FS media changer daemon designed specifically for the automated library.

NFS Network file system. A file system distributed by Sun that provides transparent access to remote file systems on heterogeneous networks.

NIS The Sun OS 4.0 (minimum) Network Information Service. A distributed network database containing key information about systems and users on the network. The NIS database is stored on the master server and all 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, such as 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. Preallocation can be specified only for a file that is size zero. For more information, see the `setfa`(1) man page.

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 Sun StorEdge SAM-FS utility that reclaims space on cartridges that is occupied by expired archive copies.

release priority The priority according to which a file in a file system is released after being archived. Release priority is calculated by multiplication of various weights of file properties and then summation of the results.

releaser A Sun StorEdge SAM-FS 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 according to high and low thresholds.

remote procedure call 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.

See also *disk striping* and *striping*.

RPC Remote procedure call. 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. See also *samfsrestore*.

SAM-QFS A configuration that combines the Sun StorEdge SAM-FS software with the Sun StorEdge QFS file system. SAM-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 Sun StorEdge SAM-FS command set as well as standard UNIX file system commands.

`samfsrestore` A program that restores inode and directory information from a control structure dump. See also *samfsdump*.

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, 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.

striped group A collection of devices within a file system that is defined in the `mcf`(4) file as one 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).

stripe size The number of disk allocation units (DAUs) to be allocated before writing proceeds to the next device of a stripe. If the `stripe=0` mount option is used, the file system uses round-robin access, not striped access.

striping A data access method in which files are simultaneously written to logical disks in an interlaced fashion. SAM-QFS file systems provide two types of striping: "hard striping," using stripe groups, and "soft striping," using the `stripe=`*x* mount parameter. Hard striping is enabled when a file system is set up, and requires the definition of stripe groups within the `mcf`(4) file. Soft striping is enabled through the `stripe=`*x* mount parameter, and can be changed for the file system or for individual files. It is disabled by setting `stripe=0`. Hard and soft striping can both be used if a file system is composed of multiple stripe groups with the same number of elements. See also *round robin*.

Sun SAM-Remote client A Sun StorEdge SAM-FS system with a client daemon that contains a number of pseudodevices, and can also have its own library devices. The client depends on a Sun SAM-Remote server for archive media for one or more archive copies.

Sun SAM-Remote server Both a full-capacity Sun StorEdge SAM-FS storage management server and a Sun SAM-Remote server daemon that defines libraries to be shared among Sun SAM-Remote clients.

superblock A data structure in the file system that defines the basic parameters of the file system. The superblock 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 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).

timer Quota software that keeps track of the period starting when a user reaches a soft limit and ending when the hard limit is 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. In the context of archiving to removable media cartridges, the VSN is a logical identifier for magnetic tape and optical disk that is written in the volume label. In the context of 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

.cshrc file, 25, 27
.inodes file, 54
.login file, 25, 27
.profile file, 25, 27

A

administrator group
  creating, 52
AMD hardware platform, 80
API
  dependent applications, 94
  libraries, 108
application programming interface, see API
archive management (Sun StorEdge SAM-FS), 2
archiver
  idling, 80
  wait mode, 80
aridle command, 80
auth_attr file, 102

B

backing up
  configuration information, 76
  files, 83
  metadata, 55
boot(1M) command, 84

C

checking the file system, 94
chmod(1) command, 43
chown(1) command, 43
clock synchronization, 58
commands, 105
configuration
  files, 101
  Sun StorEdge QFS, 30
  system logging, 52
crontab file, 55

D

daemons
  fsmgmtd, 91
  sam-rpcd, 108
  sam-sharefsd, 69
  verifying, 69
  xntpd, 58
data backups, 54
DAU
  finding, 93
  in a shared file system, 42
  specifying, 41, 115
  specifying, example, 113
defaults.conf file, 48, 76, 101
Device identifier (DID) devices, 124
device redundancy, 11
dfstab file, 45
DID devices, see Device identifier (DID) devices
directories
  created during installation, 98
disk allocation unit, see DAU
disk cache, 17

disk space
verifying, 18
du(1) command, 105
dump files, 54

E

EFI labels, 81

F

file system
commands, 107
configuration, 30
highly available, 122
shared fs example, 118
unmounting, 86
File System Manager, 108
daemon, 91
installing, 26, 89
mounting a file system, 43
requirements, 15
using, 28
format(1M) command, 14, 110, 119, 127
sample output, 119, 124
fsck(1M) command, 35
fsmadm(1M) command, 106
fsmgmtd daemon, 91
fsmgr_setup(1M) command, 106
fuser(1M) command, 86

G

group file, 52, 58
groupadd(1M) command, 52

H

HA Storage Plus, 72
hardware
AMD and SPARC platforms, 80
requirements, 7
upgrade, 79
high availability disks, 12
highly available file system, 72, 122
host systems, 58
hosts file
example, 65
fields, 63
Sun Cluster OE, 65
hosts.*fs-name* file, 63
hosts.*fsname*.local file, 66

I

initialize a file system, sammkfs(1M) command, 41
inode files, 54
inquiry.conf file, 76
installation
directories created, 98
files created, 99
instructions, 23
modified files, 101
overview, 4
requirements, 5
site-created files, 101

K

kill(1M) command, 86

L

libsam library, 108
libsamrpc library, 108
licensing, 20
general information, xviii
upgrades, 79
Linux client
EFI disk labels, 81
installing, 25
local hosts file, 66
example, 67
fields, 66
log_rotate.sh(1M) command, 53
logging, 52
ls(1) command, see sls(1) command

M

MANPATH variable, 25
mcf file, 30, 111, 113, 114, 116, 121, 122, 124
examples, 109, 122
for highly available file system, 59
for shared file systems, 59, 60
propagating changes, 33, 92
shared file system example, 61
upgrading a server, 76
verifying, 33, 91
message logging, 52

metadata
backing up, 55
dump files, 54
overview, 54
shared file systems, 119
metadata server
changing, 44
for shared file system, 7
obtaining addresses, 67
mount
parameters, 36
point, 35
using File System Manager, 43
mount(1M) command, 42, 84, 94, 107

N

Network Management Station
installing on Sun StorEdge QFS, 49
NFS sharing file systems, 45, 48

O

operating system requirements, 7
Oracle database, 53

P

packages
SUNWqfsr and SUNWqfsu, 24, 88
passwd file, 58
PATH variable, 25
pkgadd(1M) command, 24, 88, 96, 97
pkginfo(1M) command, 87
pkgrm(1M) command, 87, 88, 95
prtvtoc(1M) command, 82

Q

qfsdump(1M) command, 54, 83, 84, 107
running automatically, 55
qfsrestore(1M) command, 54, 85, 107
quotas, 107

R

redundant disks, 11
verifying, 12
release
numbering, 97
packages, 19
Release Notes, 19
remote notification
enabling, 50
facility, 49
removing software, 87
round-robin data layout, 112
rpcbind service, 90

S

sambcheck(1M) command, 107
samchaid(1M) command, 107
samcmd(1M) command, 106
samd(1M) config command, 40
samexplorer(1M) command, 106
samfs file system type, 35
samfs.cmd file, 38, 101
creating with File System Manager, 39
editing, 39
samfsck(1M) command, 94, 107
samfsconfig(1M) command, 60, 76, 107
sample output, 77
sam-fsd(1M) command, 91
samfsdump(1M) command, 107
samfsinfo(1M) command, 107
samfsrestore(1M) command, 107
samfstyp(1M) command, 107
samgrowfs(1M) command, 107
sammkfs(1M) command, 41, 107, 113, 115, 117
samncheck(1M) command, 107
SAM-QFS
definition, 2
samquota(1M) command, 107
samquotastat(1M) command, 107
sam-rpcd daemon, 108
samsharefs(1M) command, 44, 67, 107
sam-sharefsd daemon, 69
samtrace(1M) command, 107
samu(1M) command, 106, 108
samunhold(1M) command, 107
SAN attached storage
SMI VTOC8 disk labels, 81
scdidadm(1M) command, 12, 124
scrgadm(1M) command, 71, 72
scstat(1M) command, 73

scswitch(1M) command, 73
sdu(1) command, 105
server
  AMD and SPARC platforms, 80
  hardware upgrades, 79
  requirements, 7
  upgrading, 75
set_admin(1M) command, 52
setfa(1) command, 105
sfind(1) command, 106
share(1M) command, 45
shared file system, 2
  changing servers, 44
  configuration requirements, 7
  example configuration, 118
  host types, 60
  in Sun Cluster OE, 124
  Linux client, 3
  mcf file, 60
  metadata, 119
  preparing hosts, 58
  specifying a DAU, 42
  unsharing, 86
shared hosts file, 63
  example, 65
  fields, 63
  Sun Cluster OE, 65
shared resource in Sun Cluster OE, 72
Simple Network Management Protocol, see Network Management Station
sls(1) command, 106
SMI labels, converting to EFI, 82
SMI VTOC8 disk labels, 81
SNMP, see Network Management Station
software
  commands, 105
  installation, 24, 99
  overview, 1
  packages, 19, 88, 97
  release numbers, 97
  uninstalling, 87
  upgrade, 75
  upgrades (OS), 95
  user interfaces, 108
Solaris OS
  files modified during installation, 101
  patches, 6
  upgrading, 95
SPARC hardware platform, 80
squota(1) command, 106
st.conf file, 76
storage (Sun StorEdge SAM-FS), 2
stripe= option in vfstab file, 116
stripe=1 mount parameter, 36
striping example, 113, 114
Sun Cluster OE, 3
  bringing shared resource online, 72
  DID devices
  HA Storage Plus resource, 72
  highly available file system, 122
  mcf file example, 59
  requirements, 8
  shared file system, 124
  shared hosts, 60
  shared hosts file, 65
  SUNW.qfs resource type, 70
  Verifying configuration, 9
Sun StorEdge QFS
  configuration, 30
  initial installation, 23
  upgrade, 75
Sun StorEdge QFS application programming interface (API), 94, 108
Sun StorEdge SAM-FS, using with QFS, 2
Sun StorEdge Traffic Manager, 81
SUNW.qfs(5) resource, 71
SUNWqfsr and SUNWqfsu packages, 24, 88
superblocks, version 1 and 2, 83
syslog(3) interface, 52
syslog.conf file, 52
system administrator commands, 106
system files modified, 101

T

time synchronization, 58
trace_rotate(1M) command, 107

U

umount(1M) command, 86
uninstalling the software, 87
unmounting using File System Manager, 86
unshare(1M) command, 86

upgrading
hardware, 79
licenses, 79
servers, 75
software, 75
Solaris, 95
user commands, 105
user_attr file, 102
utilities, 108

V

version 1 and version 2 superblocks, 83
vfstab file, 35, 87, 92, 96
editing, 36
example, 112, 113, 114, 116
fields, 35
updating with File System Manager, 36

W

web browser requirements, File System Manager, 16

X

x64 platform, see AMD
xntpd(1M) daemon command, 58