

Sun QFS File System 5.3 Configuration and Administration Guide

Copyright © 2011, 2012, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique:

U.S. GOVERNMENT END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour ce type d'applications.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée d'The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation.

Contents

Preface	11
1 File System Overview	15
General File System Configurations	15
File System Features	15
Volume Management	16
Support for Paged and Direct I/O	16
High Capacity	16
Fast File System Recovery	17
Metadata Storage	17
vnode Interface	17
Shared File System Support	18
Additional File System Features	19
Sun QFS File Systems Design Basics	19
Inode Files and File Characteristics	19
Specifying Disk Allocation Units	20
File Allocation Methods	26
Metadata Allocation	26
Round-Robin Allocation	26
Striped Allocation	28
Striped Groups	31
Mismatched Striped Groups	34
Example of a Mismatched Striped Group	35
Per-Logical Unit Number (LUN) Allocation Control	38
2 About the Master Configuration File	39
Basic mc f File Structure	39

mcf File Fields	40
Equipment Identifier Field (Required)	40
The Equipment Number Field (Required)	42
The Equipment Type Field (Required)	42
Family Set Field (Required)	43
Device State Field (Optional)	43
Additional Parameters Field (Optional)	44
Examples of mcf Files	44
Interactions Among File Settings, Options, and Directives Introduction	45
3 mcf File Examples	47
Configuration Examples for Local File Systems	47
Simple File System Configuration Example	47
Round-Robin Configuration Example	49
Local Striping Configuration Example	50
Striped Group Configuration Example	52
Configuration Example for a Shared File System on an Oracle Solaris OS Platform	53
▼ How to Configure the Shared File System	55
Configuration Examples for Highly Available File Systems	57
▼ How to Create an mcf File for a Highly Available File System	57
Configuration Example for a Shared File System on an Oracle Solaris Cluster Platform	58
▼ How to Create an mcf File for a Shared File System in an Oracle Solaris Cluster Environment	58
4 Configuring the File System	63
Function of the mcf File	63
Initializing a File System	63
Configuration Examples	64
▼ How to Create a Round-Robin Disk Configuration	65
▼ How to Create a Striped Disk Configuration	65
▼ How to Create a Striped Group Configuration	66
5 Configuring a Shared File System	69
Using Shared QFS With NFS	69
▼ How to Configure Shared Sun QFS With NFS	69

Mounting and Unmounting Shared File Systems	70
▼ How to Mount a Shared File System	71
▼ How to Unmount a Shared File System	71
Adding or Removing a Client Host	72
▼ How to Add a Client Host to a Shared File System	72
▼ How to Remove a Client Host From a Shared File System	75
Updating the mcf file in a Shared File System Environment	77
Creating the Local Hosts Configuration File	79
Changing the Metadata Server	81
Changing the Metadata Server in a Shared File System Environment	81
Changing the Metadata Server in an Archiving Environment	83
Converting an Unshared File System to a Shared File System	84
▼ How to Convert an Unshared Metadata Server to a Shared Metadata Server	85
▼ How to Add a Client to the Metadata Server	86
Converting a Shared File System to an Unshared File System	88
▼ How to Remove a Client From a Shared File System	88
▼ How to Convert a Shared Metadata Server to an Unshared System	88
Client-Server Communications in a Shared File System	89
Adding Disk Cache to a File System	90
▼ How to Add Disk Cache to a File System	91
Recreating a File System	91
▼ How to Back Up and Re-Create a File System	92
6 Administering File System Quotas	95
About File System Quotas	95
Types of Quotas, Quota Files, and Quota Records	95
Soft Limits and Hard Limits	96
Disk Blocks and File Limits	97
Enabling Quotas	97
Guidelines for Setting Up Quotas	97
▼ How to Configure a New File System to Use Quotas	98
▼ How to Configure an Existing File System to Use Quotas	99
▼ How to Assign Admin Set IDs to Directories and Files	101
Setting Infinite Quotas	101
Enabling Default Quota Values	102

Enabling Limits	102
▼ How to Enable or Change Limits for Users, Groups, or Admin Sets Using an Existing Quota File	103
Checking Quotas	104
Changing and Removing Quotas	105
▼ How to Change the Grace Period	106
Changing the Grace Period Expiration	107
Inhibiting Additional File System Resource Allocations	108
▼ How to Inhibit Additional File System Resource Allocations	109
▼ How to Remove the Quotas for a File System	110
▼ How to Correct Quotas	111
7 Advanced File System Topics	113
Using Daemons, Processes, and Tracing	113
Daemons and Processes	113
Trace Files	115
Using the <code>setfa</code> Command to Set File Attributes	117
Selecting File Attributes for Files and Directories	117
Preallocating File Space	118
Selecting a File Allocation Method and Stripe Width	118
Selecting a Striped Group Device	119
Accommodating Large Files	119
Configuring a Multireader File System	120
About I/O Types	121
Paged I/O	121
Direct I/O	121
I/O Switching	122
8 SMB Service in SAM-QFS	123
Getting Started With the Oracle Solaris SMB Service	124
<code>share</code> Command	124
Case-Insensitivity Support	124
DOS Attributes Support	125
Using ACLs to Protect Sun QFS and SAM-QFS Files	126
Mapping User and Group Identities in SAM-QFS	127

Creating and Converting File Systems to Support NFSv4 ACLs	127
Syntax Descriptions for Setting ACLs	128
ACL Inheritance	131
9 Configuring WORM-FS File Systems	135
About WORM-FS File Systems	135
Using WORM-FS With NFS Clients	136
Enabling the WORM-FS Feature	136
WORM Lite Options	138
Creating WORM Files	138
Setting the Default Retention Period	144
Setting the Retention Period Using touch	145
Using s1s to View WORM-FS Files	146
Using sfind to Find WORM-FS Files	148
10 Tunable Parameters	149
Increasing File Transfer Performance for Large Files	149
▼ How to Increase File Transfer Performance	149
Enabling Qwrite Capability	152
Setting the Write Throttle	153
Setting the Flush-Behind Rate	153
Tuning the Number of Inodes and the Inode Hash Table	154
ninode Parameter	154
nhino Parameter	154
When to Set the ninode and nhino Parameters	155
11 Using QFS File Systems with SANergy (SAN-QFS)	157
Using the SAN-QFS File System in a Heterogeneous Computing Environment	157
SAN-QFS Shared File System and Sun QFS Shared File System Comparison	158
Enabling the SAN-QFS File System	159
Before You Begin	159
▼ How to Enable the SAN-QFS File System on the Metadata Controller	160
▼ How to Enable the SAN-QFS File System on the Clients	161
▼ How to Install the SANergy Software on the Clients	161

Unmounting the SAN-QFS File System	162
▼ How to Unmount the SAN-QFS File System on the SANergy Clients	162
▼ How to Unmount the SAN-QFS File System on the Metadata Controller	162
▼ How to Unmount the SAN-QFS File System on the Sun QFS Clients	163
▼ How to Unmount the SAN-QFS File System on the Sun QFS Server	163
Troubleshooting: Unmounting a SAN-QFS File System With SANergy File Holds	163
Using <code>samgrowfs</code> to Expand SAN-QFS File Systems	164
12 Mount Options in a Shared File System	165
Shared File System Mount Options	165
Mounting in the Background: (<code>bg</code> Option)	165
Reattempting a File System Mount: (<code>retry</code> Option)	165
Declaring a Sun QFS Shared File System: (<code>shared</code> Option)	166
Tuning Allocation Sizes: (<code>minallopsz</code> and <code>maxallopsz</code> Options)	166
Using Leases in a Sun QFS Shared File System: (<code>rdlease</code> , <code>wrlease</code> , and <code>aplease</code> Options)	166
Enabling Multiple Host Reads and Writes: (<code>mh_write</code> Option)	167
Setting the Minimum Number of Concurrent Threads: (<code>min_pool</code> Option)	169
Retaining Cached Attributes: (<code>meta_timeo</code> Option)	169
Specifying Striped Allocation: (<code>stripe</code> Option)	169
Specifying the Frequency With Which Metadata Is Written: (<code>sync_meta</code> Option)	169
Enabling WORM Functionality: (<code>worm_capable</code> and <code>def_retention</code> Options)	170
13 Using the <code>samu</code> Operator Utility	171
<code>samu</code> Operator Utility at a Glance	171
Overview of the <code>samu</code> Operator Utility	176
▼ How to Start the <code>samu</code> Utility	176
▼ How to Display a <code>samu</code> Screen	177
▼ How to Stop <code>samu</code>	177
Interacting With <code>samu</code>	177
Identifying a Device	178
Accessing Online Help	178
Operator Displays	178
(a) - Archiver Status Display	179
(c) - Device Configuration Display	180

(C) - Memory Display	182
(d) - Daemon Trace Controls Display	183
(D) - Disk Volume Dictionary	184
(f) - File Systems Display	185
(F) - Optical Disc Label Display	187
(h) - Help Display	187
(I) - Inode Display	188
(J) - Preview Shared Memory Display	189
(K) - Kernel Statistics Display	190
(l) - Usage Display	191
(L) - Shared Memory Tables	191
(m) - Mass Storage Status Display	192
(M) - Shared Memory Display	194
(n) - Staging Status Display	195
(N) - File System Parameters Display	195
(o) - Optical Disc Status Display	196
(p) - Removable Media Load Requests Display	197
(P) - Active Services Display	199
(r) - Removable Media Status Display	200
(R) - Sun SAM-Remote Information Display	201
(s) - Device Status Display	201
(S) - Sector Data Display	202
(t) - Tape Drive Status Display	203
(T) - SCSI Sense Data Display	204
(u) - Staging Queue Display	205
(U) - Device Table Display	206
(v) - Automated Library Catalog Display	207
(w) - Pending Stage Queue	210
Status Codes	212
States of an Operator Display Device	214
▼ How to Change a Drive State From down to on	215
Operator Commands	215
Device Commands	216
File System Commands: I/O Management	216
File System Commands: Direct I/O Management	219
File System Commands: Sun QFS Shared File Systems	221

File System Commands: Miscellaneous	222
Miscellaneous Commands	226

Preface

The *Sun QFS Configuration and Administration Guide* provides information about managing the Sun QFS file systems.

Who Should Use This Book

This guide is intended for system administrators who are interested in managing Sun QFS file systems.

Before You Read This Book

Read *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide* before you begin to manage the Sun QFS file systems.

Related Books

- *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide*
- *Sun Storage Archive Manager 5.3 Configuration and Administration Guide*
- *Sun QFS and Sun Storage Archive Manager 5.3 Release Notes*
- *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*

Related Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Oracle is not responsible for the availability of third-party web sites mentioned in this document. Oracle 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. Oracle will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P-1 Typographic Conventions

Typeface	Description	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name% you have mail.</code>
AaBbCc123	What you type, contrasted with onscreen computer output	<code>machine_name% su</code> Password:
<i>aabbcc123</i>	Placeholder: replace with a real name or value	The command to remove a file is <code>rm filename</code> .
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . <i>A cache</i> is a copy that is stored locally. Do <i>not</i> save the file. Note: Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX system prompt and superuser prompt for shells that are included in the Oracle Solaris OS. Note that the default system prompt that is displayed in command examples varies, depending on the Oracle Solaris release.

TABLE P-2 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	\$
Bash shell, Korn shell, and Bourne shell for superuser	#
C shell	machine_name%
C shell for superuser	machine_name#

File System Overview

The Sun QFS file system is a configurable file system that presents a standard UNIX file system interface to users.

General File System Configurations

The Sun QFS file system can be used in several different configurations:

- Configured as a *stand-alone* file system on a single host
- Configured as a *shared* file system, in which multiple hosts can write to and read from the file system
- Configured as a *multireader* file system, in which only one host can write to the file system, but multiple hosts can read from the file system
- Configured with the Sun Storage Archive Manager (SAM-QFS) product to support *archiving* features
- Configured with the Oracle Solaris Cluster product for failover and *high-availability*

Unless otherwise noted, file system information throughout this document applies to archiving and non-archiving configurations.

The Sun QFS file system does not require changes to user programs or to the UNIX kernel.

File System Features

Key features of the Sun QFS file systems are described in the following sections:

- “Volume Management” on page 16
- “Support for Paged and Direct I/O” on page 16
- “High Capacity” on page 16

- “Fast File System Recovery” on page 17
- “Metadata Storage” on page 17
- “vnode Interface” on page 17
- “Shared File System Support” on page 18

Volume Management

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

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

Support for Paged and Direct I/O

The Sun QFS file system supports two different types of I/O:

- Paged I/O (also known as cached or buffered I/O) – When *paged I/O* is used, user data is cached in virtual memory pages and the kernel writes the data to disk. The standard Oracle Solaris OS interfaces manage paged I/O. This is the default type of I/O for Sun QFS.
- Direct I/O – When *direct I/O* is used, user data is written directly from user memory to disk. You can specify direct I/O by using the Oracle Solaris OS `directio(3C)` function call or the `setfa` command with the `-D` option. By using direct I/O, you can realize substantial performance improvements for large block, sequential, aligned I/O.

High Capacity

The Sun QFS software supports files of up to 2^{63} bytes in length. Very large files can be striped across many disks or RAID devices, even within a single file system, because Sun QFS file systems use true 64-bit addressing. In contrast, standard UNIX file systems are not true 64-bit file systems.

The number of file systems that you can configure is virtually unlimited. The volume manager enables each file system to include up to 252 device partitions, typically disk partitions. Each partition can include up to 16 terabytes of data. This configuration offers virtually unlimited storage capacity.

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

In a Sun QFS file system, the inodes are located on the metadata devices and can be separated from the file data devices. In practice, the size of your metadata (`mm`) devices limits the number of files in a Sun QFS file system, but you can increase the maximum number of files by adding more metadata devices. The hard limit on the number of files is $2^{32} - 1$ files, and the recommended limit is 10^8 files.

Fast File System Recovery

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

A Sun QFS file system often does not require a file system check after a disruption that prevents the file system from being written to disk using the `sync` command. In addition, Sun QFS file systems recover from system failures without using journaling. They recover dynamically by using identification records, serial writes, and error checking for all critical I/O operations. After a system failure, even multiterabyte-sized Sun QFS file systems can be remounted immediately.

Metadata Storage

File systems use metadata to reference file and directory information. Typically, metadata resides on the same device as the file data. However, the Sun QFS file system has the option of separating the file system metadata from the file data by storing them on separate devices. The Sun QFS file system enables you to define one or more separate metadata devices in order to reduce device head movement and rotational latency, improve RAID cache utilization, or mirror metadata without mirroring file data.

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

vnode Interface

The Sun QFS file system is implemented through the standard Oracle Solaris OS virtual file system (`vfs/vnode`) interface.

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

The kernel intercepts all requests for files, including those that reside in Sun QFS file systems. If a file is identified as a Sun QFS file, the kernel passes the request to the appropriate file system for handling. Sun QFS file systems are identified as type `samfs` in the `/etc/vfstab` file and through the `mount` command.

Shared File System Support

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

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

The shared file system can be implemented either as a Sun QFS shared file system or as a SAM-QFS shared file system. It can use either `ms` or `ma` file system types.

Sun QFS shared file systems do not support the following:

- Certain file types:
 - `b`– Block special files
 - `c`– Character special files
 - `p`– FIFO (named pipe) special files
- Segmented files. You cannot implement a SAM-QFS shared file system in a segmented-file environment.
- Mandatory locks. An `EACCES` error is returned if the mandatory lock is set. Advisory locks are supported, however. For more information about advisory locks, see the `fcntl(2)` man page.

For more information about shared file systems, see [Chapter 5, “Configuring a Shared File System.”](#)

Additional File System Features

The following additional features are also supported by the Sun QFS file system:

- Application programming interface (API) routines – API routines enable a program to perform various specialized functions, such as preallocating contiguous disk space or accessing a specific striped group. For more information about these routines, see “[intro_libsam\(3\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.
- Adjustable disk allocation units (DAUs) – The DAU is the basic unit of online storage. The Sun QFS file system software includes an adjustable DAU, which is useful for tuning file systems with the physical disk storage device and for eliminating the system overhead caused by read-modify-write operations. For information about adjusting the DAU size, see “[Specifying Disk Allocation Units](#)” on page 20.
- Support for multiple striped groups – To support multiple RAID devices in a single file system, Sun QFS software supports the definition of striped groups. You can optimize disk block allocation for a striped group, thereby reducing the overhead for updating the on-disk allocation map. Users can assign a file to a striped group either through an API routine or by using the “[setfa\(1\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

Sun QFS File Systems Design Basics

Sun QFS file systems are multi threaded, advanced storage management systems. To take maximum advantage of the software’s capabilities, create multiple file systems whenever possible.

Sun QFS file systems use a linear search method for directory lookups, searching from the beginning of the directory to the end. As the number of files in a directory increases, the search time through the directory also increases. Search times can become excessive when you have directories with thousands of files. These long search times are also evident when you restore a file system. To increase performance and speed up file system dumps and restores, keep the number of files in a directory under 10,000.

The directory name lookup cache (DNLC) feature improves file system performance. This cache stores the directory lookup information for files whose paths are short (30 characters or less), removing the need for directory lookups to be performed on the fly.

Inode Files and File Characteristics

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

Inodes are stored in the `.inodes` file located under the file system mount point. Like a standard Oracle Solaris OS inode, a Sun QFS file system inode contains the file's POSIX standard inode times: file access, file modification, and inode changed times. A Sun QFS file system inode includes other times as well, as shown in the following table.

TABLE 1-1 Content of `.inode` Files

Time	Incident
access	Time the file was last accessed. POSIX standard.
modification	Time the file was last modified. POSIX standard.
changed	Time the inode information was last changed. POSIX standard.
attributes	Time the attributes specific to the file system were last changed.
creation	Time the file was created.
residence	Time the file changed from offline to online or vice versa.

Note – If the WORM-FS (write-once read-many) package is installed, the inode also includes a retention-end date. See [Chapter 9, “Configuring WORM-FS File Systems,”](#) for more information.

Specifying Disk Allocation Units

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

DAU Settings and File System Geometry

Sun QFS file systems use an adjustable DAU. You can configure the DAU to tune the file system to the physical disk storage device. This feature minimizes the system overhead caused by read-modify-write operations and is therefore particularly useful for applications that manipulate very large files. For information about how to control the read-modify-write operation, see [“Increasing File Transfer Performance for Large Files”](#) on page 149.

Each file system can have its own unique DAU setting even if it is one of several mounted file systems active on a server. The possible DAU settings differ depending on the type of file system you are using. The DAU setting is determined through “[sammkfs\(1M\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual* when the file system is created. It cannot be changed dynamically.

DAU settings work in conjunction with the device and file system definitions specified in the master configuration (`mcf`) file. For details about the `mcf` file, see [Chapter 2, “About the Master Configuration File,”](#) and “[mcf\(4\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

ms and ma File Systems

Two file allocation schemes are available to you:

- An `ms` file system type – File system data and file system metadata are on the same device
- An `ma` file system type – File system data and file system metadata are on different devices

For a simple Sun QFS file system, such as one on a single partition, the file system is defined in your `mcf` file by an Equipment Type value of `ms`. In the `ms` file system, the only device type allowed is type `md`, and both metadata and file data are written to the `md` devices. By default, the DAU on an `md` device is 64 kilobytes.

A more complex Sun QFS file system installed on multiple partitions is defined as Equipment Type `ma` in your `mcf` file. In an `ma` file system, metadata is written to `mm` devices, and data can be written to `md`, `mr`, or `g XXX` devices.

Within an `ma` file system you can mix devices as follows:

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

For more information about these device types, see [Chapter 2, “About the Master Configuration File.”](#)

Dual and Single Allocation Schemes

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

- On `md` data devices, the small allocation is 4 kilobytes, and the large allocation is a DAU. The default DAU is 64 kilobytes. You can override this default when the file system is initialized by using the `-allocation-unit` option to the `sammkfs` command. The DAU size can be 16, 32, or 64 kilobytes.

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

Note – When using an `ms` file system, the stripe width should be set to greater than zero to stripe metadata information across the disk. However, you should read and understand “[Stripe Widths on Data Disks](#)” on page 23 before setting the stripe width and DAU size.

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

Depending on the type of file data stored in the file system, a larger DAU size can improve file system performance significantly. For information about tuning file system performance, see [Chapter 7, “Advanced File System Topics.”](#)

Only `ma` Sun QFS file systems can include devices that use a single allocation scheme. These file systems consist of separate metadata devices and data devices, as follows:

- The metadata devices can be defined only as Equipment Type `mm`.
- The data devices can be defined as Equipment Type `md`, `mr`, or `g XXX`. The `md` devices are limited to DAU sizes of 16 kilobytes, 32 kilobytes, or 64 kilobytes.

The `mr` and `g XXX` devices follow a single allocation scheme. You can mix `mr` and `g XXX` devices in a file system, but you cannot mix `md` devices with either `mr` or `g XXX` devices in a file system. The `mr` and `gXXX` devices can be set to a minimum DAU allocation of 8 kilobytes for devices in an `ma` file system. This setting is optimal for workloads with the majority of file sizes at or below 8 kilobytes.

The DAU size for file systems that use `mr` and `g XXX` data devices is configurable. The possible DAU sizes that can be used on data devices depend on the Equipment Type value assigned to each data device in the `mc f` file. The following table shows these DAU sizes.

TABLE 1-2 Equipment Type Values and DAU Sizes

Equipment Type	DAU Sizes
<code>mr</code> or <code>gXXX</code>	You can specify different DAU sizes by adjusting the default size in 8-kilobyte increments. The DAU size can be from 8 kilobytes to 65,528 kilobytes (64 megabytes). The default DAU size is 64 kilobytes for <code>mr</code> or 256 kilobytes for <code>gXXX</code> .

TABLE 1-2 Equipment Type Values and DAU Sizes (Continued)

Equipment Type	DAU Sizes
md	This type of device uses a dual allocation scheme. The DAU can be configured to be 16, 32, or 64 kilobytes in length. The default DAU size is 64 kilobytes. An md device in an ma file system is used to store data only, not metadata. An md device in an ms file system is used to store both file data and metadata.

Note – If you created your file system using version 3.5 of the software, or built it using the `sammkfs` compatibility mode flag in version 4 of the software, you might be using a version 1 superblock. In the version 1 superblock, mm devices do not use the dual allocation scheme, and the allocation for mm devices is 16 kilobytes. Only a version 2 superblock enables you to define md devices in a Sun QFS file system. To find out whether you are using a version 1 superblock, use the `samfsinfo` command.

Data Alignment

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

$$\text{allocation-unit} = \text{RAID-stripe-width} \times \text{number-of-data-disks}$$

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

Data files are allocated as striped or round-robin through each striped group (gXXX) or data disk (mr or md) within the same file system.

A mismatched alignment hurts performance because it can cause a read-modify-write operation.

Stripe Widths on Data Disks

Stripe width defaults differ between ms and ma file systems. The stripe width is specified by the `-o stripe= n` option in the mount command. If the stripe width is set to 0, round-robin allocation is used.

Stripe Widths on ms File Systems

On ms file systems, the stripe width is set at mount time. The following table shows default stripe widths.

TABLE 1-3 ms File System Default Stripe Widths

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

For example, if you run `sammkfs` command with default settings, the default large DAU is 64 kilobytes. If no stripe width is specified when the `mount` command is issued, the default is used, and the stripe width set at mount time is 2.

Note –

- To stripe metadata information across the disk in an `ms` file system, set the stripe width to greater than zero.
 - If you multiply the number in the first column of [Table 1-3](#) by the number in the second column, the resulting number is 128 kilobytes. Sun QFS file systems operate most efficiently if the amount of data being written to disk is at least 128 kilobytes.
-

Stripe Widths on `ma` File Systems Not Using Striped Groups

On `ma` file systems, the stripe width that is set at mount time depends on whether striped groups are configured. A striped group is a collection of devices that are striped as a group. For more information about striped groups, see [“File Allocation Methods” on page 26](#). This section describes stripe widths for Sun QFS file systems that are configured without stripe groups.

If striped groups are not configured, the DAU and stripe width relationships on `ma` file systems are similar to those for `ms` file systems. The difference is that DAUs larger than 64 kilobytes are possible and that the DAU is configurable in 8-kilobyte blocks. The maximum DAU size is 65,528 kilobytes.

By default, if no stripe width is specified, the amount of data written to disk is at or near 128 kilobytes. Sun QFS file systems are most efficient if write operations write at least one whole stripe per I/O request. The following table shows the default stripe widths.

TABLE 1-4 Default Stripe Widths for `ma` File Systems Not Using Striped Groups

DAU	Default Stripe Width	Amount of Data Written to Disk
16 kilobytes	8 DAUs	128 kilobytes
24 kilobytes	5 DAUs	120 kilobytes
32 kilobytes	4 DAUs	128 kilobytes

TABLE 1-4 Default Stripe Widths for ma File Systems Not Using Striped Groups *(Continued)*

DAU	Default Stripe Width	Amount of Data Written to Disk
40 kilobytes	3 DAUs	120 kilobytes
48 kilobytes	2 DAUs	96 kilobytes
56 kilobytes	2 DAUs	112 kilobytes
64 kilobytes	2 DAUs	128 kilobytes
72 kilobytes	1 DAU	72 kilobytes
128 kilobytes	1 DAU	128 kilobytes
> 128 kilobytes	1 DAU	DAU size

Stripe Widths on ma File Systems Using Striped Groups

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

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

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

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

For more information, see [“File Allocation Methods” on page 26](#).

Stripe Widths on Metadata Disks

You can use the `-o mm_stripe=n` option to the `mount_samfs` command to stripe metadata information on the metadata disk. The default stripe width is `-o mm_stripe=1`, which specifies that the file system write one 16-kilobyte DAU to a metadata disk before switching to the next metadata disk. The small 4-kilobyte DAU is used for metadata disks.

By default, if you have multiple metadata devices, metadata is allocated as specified in the `-o mm_stripe=n` option to the `mount` command. The setting can be as low as `-o mm_stripe=0`, which disables striping, or as high as `-o mm_stripe=255`.

File Allocation Methods

The Sun QFS software enables you to specify both round-robin and striped allocation methods. The following table shows the default file allocation methods used.

TABLE 1-5 Default Allocation Methods

File System	Metadata	File Data
Stand-alone file system	Striped	Striped
Shared file system	Striped	Round-robin
Striped groups	Striped	Round-robin

Metadata Allocation

Metadata allocation varies according to the type of file system:

- For *ms* file systems, metadata is allocated across the *md* devices.
- For *ma* file systems, metadata is allocated across the *mm* devices. No file data is allocated on the *mm* devices.

Inodes are 512 bytes in length. Directories are initially 4 kilobytes in length. The following table shows how the system allocates metadata.

TABLE 1-6 Metadata Allocation

Metadata Type	Allocation Increments for <i>ma</i> File Systems	Allocation Increments for <i>ms</i> File Systems
Inodes (. <i>inodes</i> file)	16-kilobyte DAU	16-kilobyte, 32-kilobyte, or 64-kilobyte DAU
Indirect blocks	16-kilobyte DAU	16-kilobyte, 32-kilobyte, or 64-kilobyte DAU
Directories	4-kilobyte blocks and 16-kilobyte DAUs	4 kilobytes, up to a 32-kilobyte total, than DAU size

Round-Robin Allocation

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

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

To specify round-robin allocation explicitly, include `stripe=0` in the `/etc/vfstab` file.

The following figures depict round-robin allocations in `ms` and `ma` file systems. In both figures, file 1 is written to disk 1, file 2 is written to disk 2, file 3 is written to disk 3, and so on. When file 6 is created, it is written to disk 1, restarting the round-robin allocation scheme.

FIGURE 1-1 Round-Robin Allocation in an `ms` File System Using Five Devices

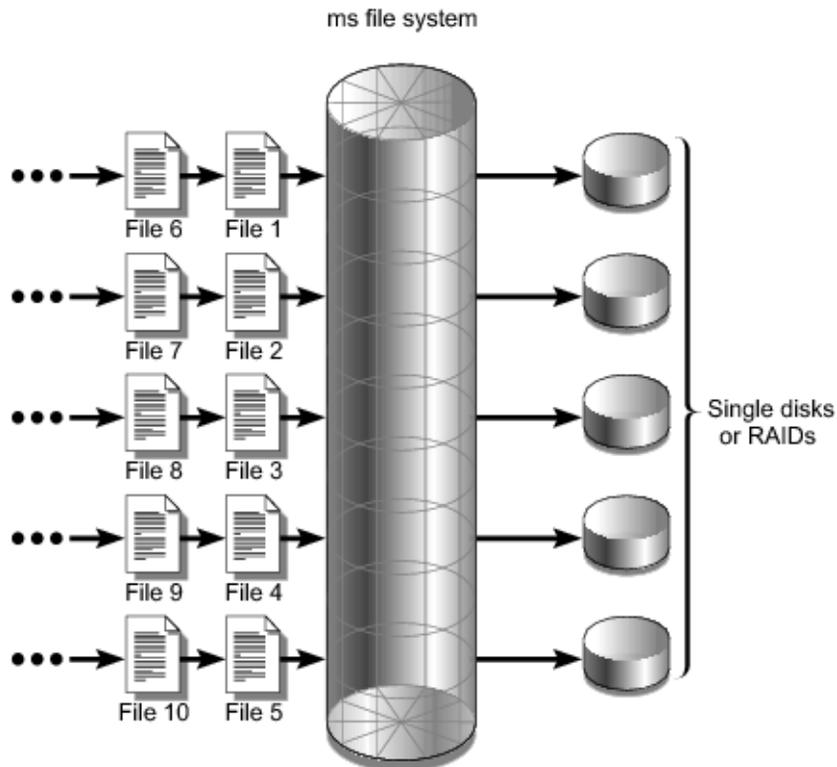
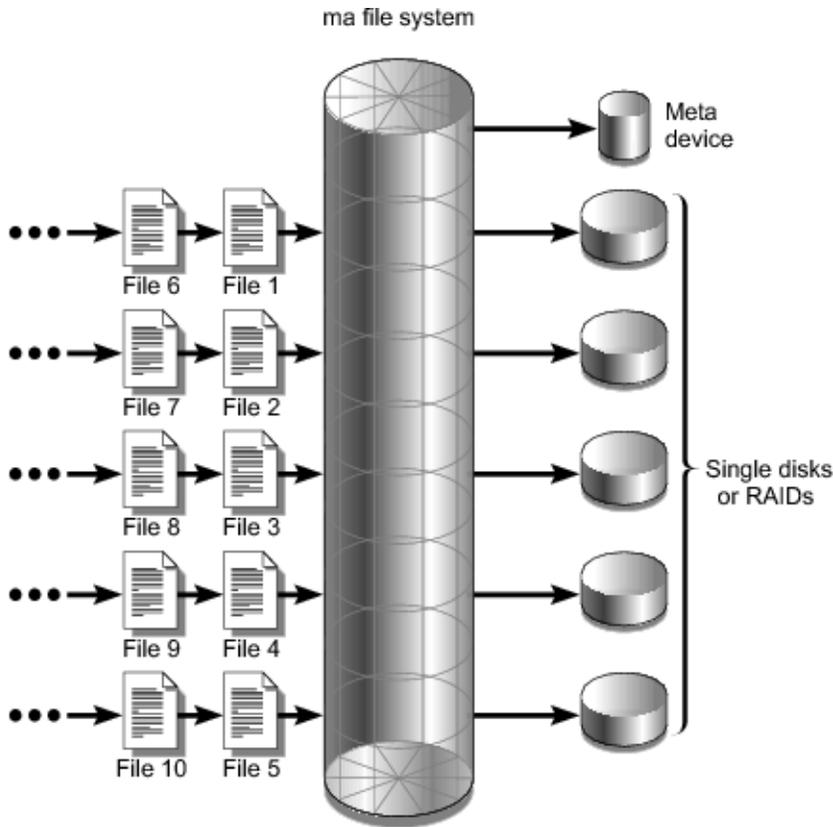


FIGURE 1-2 Round-Robin Allocation in an ma File System Using Five Devices



Striped Allocation

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

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

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

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

When a file system writes a file to an `mr` device, it writes first to one DAU, then to another, and so on. The `mr` devices have only one DAU size.

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

The following figures depict `ms` and `ma` file systems that use striped allocations. In these figures, `DAU × stripe-width` bytes of the file are written to disk 1, `DAU × stripe-width` bytes of the file are written to disk 2, and so on. The order of the stripe is first-in-first-out for the files. Striping spreads the I/O load over all the disks.

FIGURE 1-3 Striping in an ms File System Using Five Devices

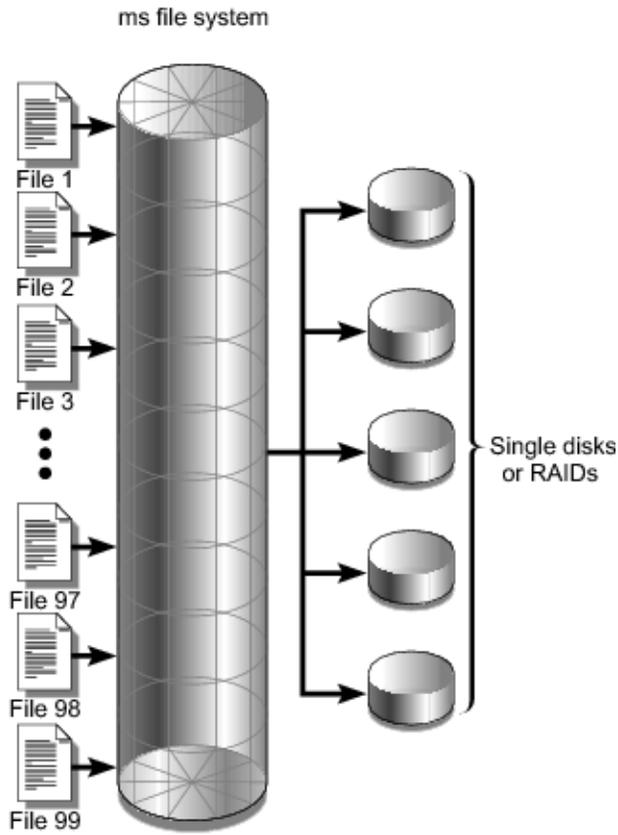
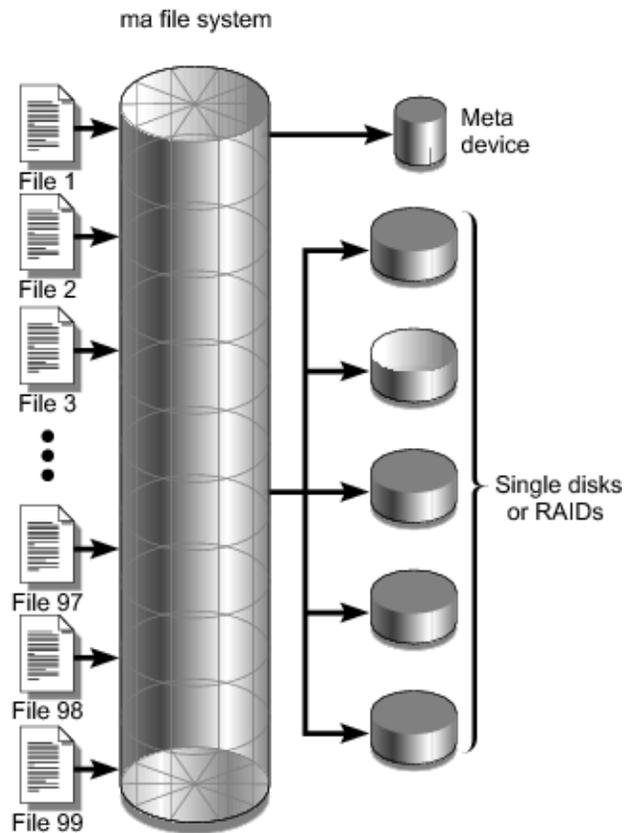


FIGURE 1-4 Striping in an `ma` File System Using Five Devices

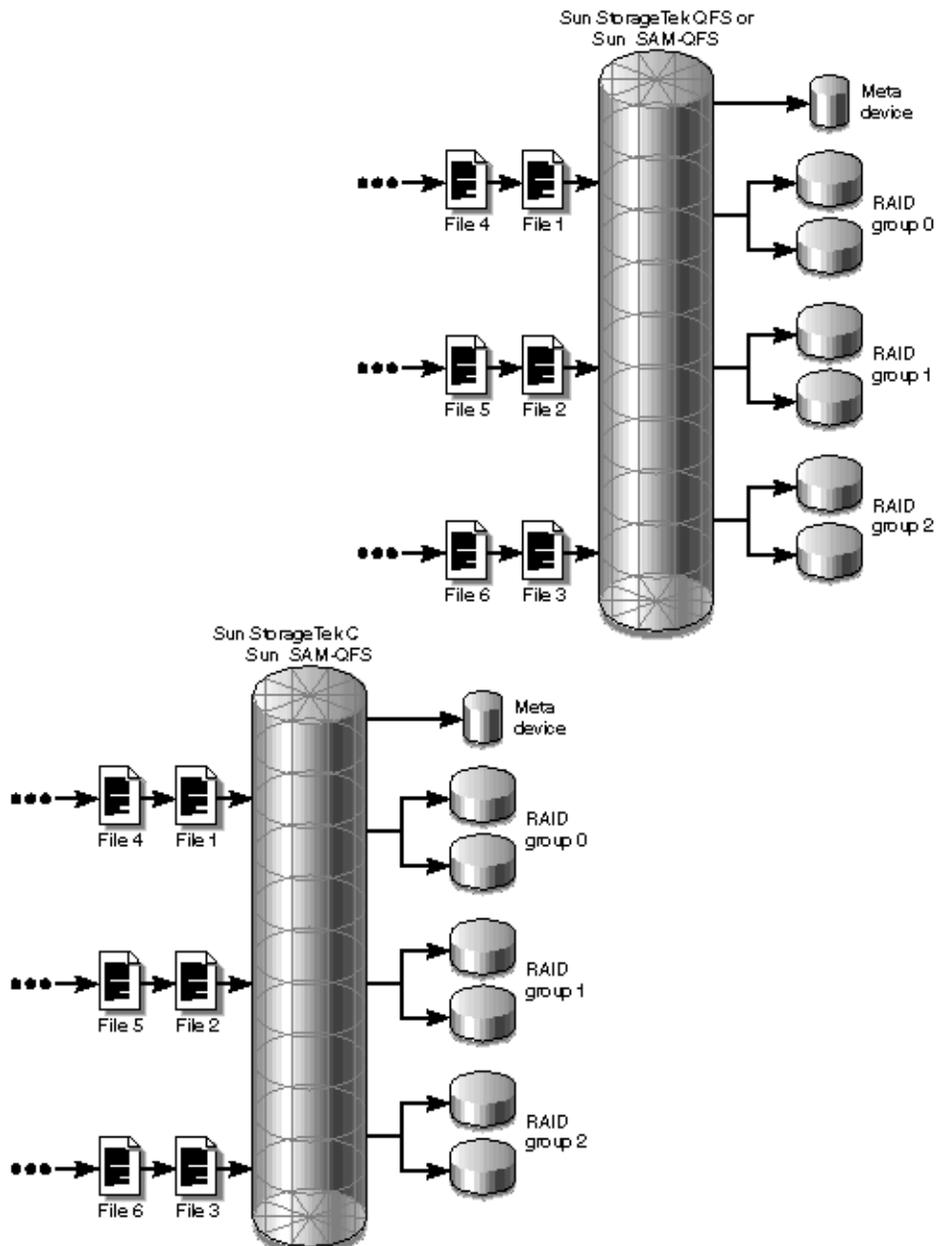
Striped Groups

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

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

The following figure depicts an `ma` file system using striped groups and a round-robin allocation. In this figure, files written to the `qfs1` file system are allocated round-robin among the defined striped groups `g0`, `g1`, and `g2`. Each group consists of two physical RAID devices.

FIGURE 1-5 Sun QFS Round-Robin Striped Groups



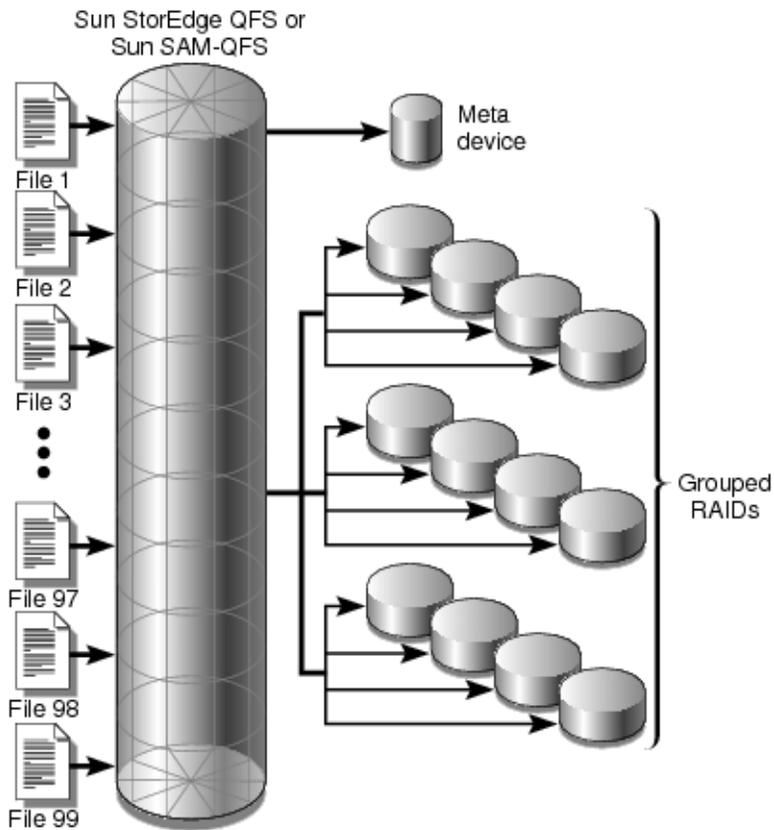
For the configuration in the figure, the mount point option in `/etc/vfstab` is set to `stripe=0`. The following example shows the `mcf` file that declares these striped groups.

EXAMPLE 1-1 `mcf` File Showing Striped Groups

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

The following figure depicts a Sun QFS `ma` file system using striped groups and striped allocation. Files written to the `qfs1` file system are striped through groups `g0`, `g1`, and `g2`. Each group includes four physical RAID devices. The mount point option in `/etc/vfstab` is set to `stripe=1` or greater.

FIGURE 1-6 Sun QFS Striped Group Allocation



Mismatched Striped Groups

You can build a file system that has mismatched striped groups, which are multiple striped groups with different numbers of devices in each group. Sun QFS file systems support mismatched striped groups, but they do not support striping on mismatched groups. File systems with mismatched striped groups are mounted as round-robin file systems.

Note – In a file system that contains mismatched striped groups, a single file cannot span multiple stripe groups. If the stripe group on which the file resides fills, it cannot be extended. If mismatched stripe groups are present, use the `setfa` command's `-g` option to direct files into the desired group. For more information, see “[setfa\(1\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

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

The example in the next section shows how a file system can be set up with mismatched striped groups to store different types of files.

Example of a Mismatched Striped Group

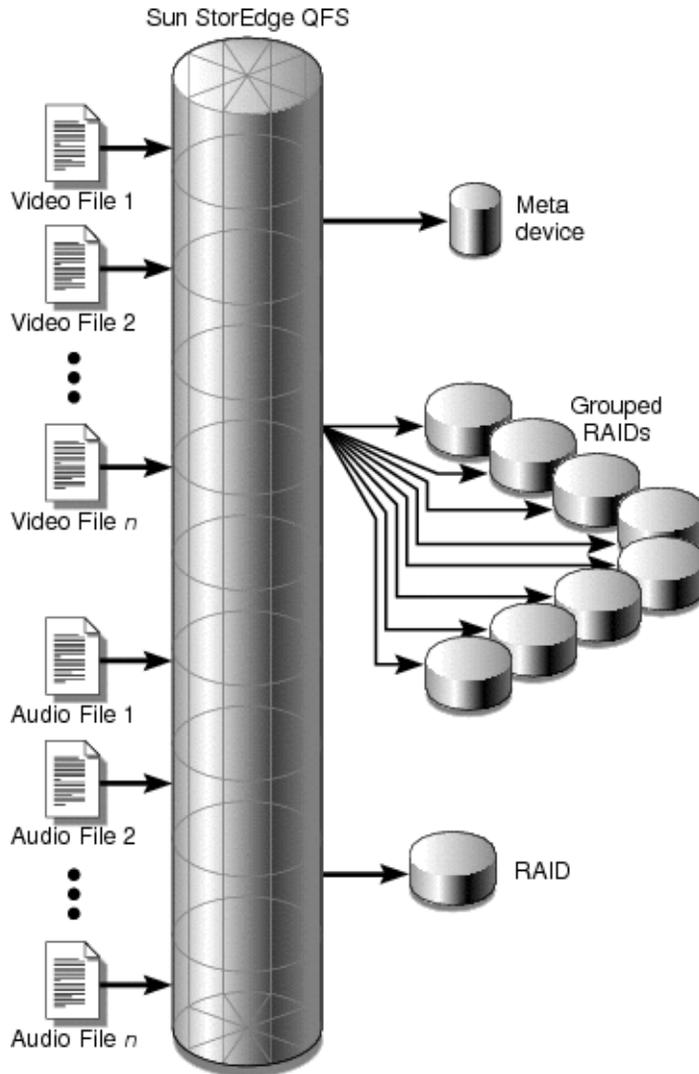
Suppose that you need to create a file system at your site that contains both video and audio data.

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

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

The following figure depicts the file system needed. It is an `ma` file system that uses mismatched striped groups in a striped allocation.

FIGURE 1-7 Sun QFS File System Using Mismatched Striped Groups in a Striped Allocation



The following table shows the characteristics of this sample file system.

TABLE 1-7 Sample File System Characteristics

Characteristics	Notes
File system name	avfs

TABLE 1-7 Sample File System Characteristics (Continued)

Characteristics	Notes
Number of stripe groups	Two. The video file group is g0. The audio file group is g1.
Stripe width	0
DAU	128 kilobytes
Number of disks for g0	Eight
Minimum block size for g0	Eight disks x 128-kilobyte DAU = 1024 kilobytes. This is the amount of data written in one block write. Each disk receives 128 kilobytes of data, so the total amount written to all disks at one time is 1024 kilobytes.
Number of disks for g1	One
Minimum block size for g1	One disk x 128-kilobyte DAU = 128 kilobytes

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

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

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

The following example shows the `mcf` file for file system `avfs`.

EXAMPLE 1-2 `mcf` File for File System `avfs`

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

When the `mcf` file for this file system is ready, you can issue the `sammkfs` and `mount` commands shown in the following example to create and mount the `avfs` file system.

EXAMPLE 1-3 Commands to Create and Mount File System avfs

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

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

EXAMPLE 1-4 Commands to Create Directories in File System avfs

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

Once the directories are created, you can use the `setfa` commands shown in the following example to assign the large striped group to `video` and the small striped group to `audio`. Files created in these directories are allocated on their respective striped groups because attributes are inherited.

EXAMPLE 1-5 Commands to Set File Attributes

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

For more information about the `sammkfs` command, see “[sammkfs\(1M\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*. For more information about the `mount` commands, see “[mount_samfs\(1M\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*. For more information about the `setfa` command, see “[setfa\(1\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

Per-Logical Unit Number (LUN) Allocation Control

If necessary, you can disable allocation to a specific Sun QFS data partition by using the `naLloc` command, which prohibits any future allocation to that device. The feature is currently only usable on data partitions, not on metadata partitions.

Allocation to a partition can be restarted by either an `aLloc` or `on` command.

The allocation state of a partition (`aLlocFlag`) is persistent across boots.

The `naLloc` and `aLloc` commands are available in the `samu` interface, and the `samu on` command also sets allocation to `on`. The `samu` screens display the `naLloc` state for partitions that have been disabled. The `samtrace` and `samfsinfo` output also include the allocation state.

For more information about the `samu` interface, see [Chapter 13, “Using the samu Operator Utility.”](#)

About the Master Configuration File

The master configuration file `/etc/opt/SUNWsamfs/mcf` describes all devices that are under the control of, or used by, the Sun QFS or SAM-QFS software. When you create this ASCII file at system configuration time, you declare attributes for each device, and you group the devices in each file system into family sets.

The `mcf` file contains the information that these file systems need in order to identify and organize RAID and disk devices into file systems. It also contains entries for each automated library or device included in a file system. A sample `mcf` file is located in `/opt/SUNWsamfs/examples/mcf`.

Basic `mcf` File Structure

An `mcf` file consists of lines of specification code divided into six columns, or fields, as shown in the following example.

Equipment Identifier	Equipment Number	Equipment Type	Family Set	Device State	Additional Parameters
----------------------	------------------	----------------	------------	--------------	-----------------------

Follow these rules when entering data in the `mcf` file:

- Use either space or tab characters between the fields in the file.
- Start comment lines with a hash character (#).
- Some fields are optional. Use a dash character (-) to indicate that an optional field contains no meaningful information.

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

You can also use SAM-QFS Manager to automatically create an `mcf` file. For information about installing SAM-QFS Manager, see [Chapter 6, “Installing and Configuring SAM-QFS Manager,” in *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide*](#). For information about using SAM-QFS Manager, see its online help.

mcf File Fields

This section describes the fields in an mcf file.

Equipment Identifier Field (Required)

The Equipment Identifier field identifies the physical file system device or removable media device. If this field contains the name of a file system, it is limited to 31 characters. For all other content, this field is limited to 127 characters.

Use the Equipment Identifier field to provide the information in the following table.

Information	Identifier Length	Description
File system name	31 characters	<p>The file system name must be identical to the name in the Family Set field, and the subsequent lines in the mcf file must define all the disks or devices included in the file system. More than one file system can be declared in an mcf file. Typically, the first data line in an mcf file declares the first file system, and subsequent lines specify the devices included in the file system. Other file systems declared in the mcf file can be preceded by a blank comment line for readability.</p> <p>Note – File system names must start with an alphabetic character and can contain only alphabetic characters, numeric characters, or underscore (_) characters.</p>

Information	Identifier Length	Description
nodev keyword	127 characters	<p>The keyword nodev indicates that the system on which the mcf file resides is being used as a client host in a shared file system on an Oracle Solaris host. This keyword can appear in this field only as the Equipment Identifier for one or more metadata devices that reside on the metadata server. For more information about creating an mcf file for the members of a shared file system, see Chapter 5, “Configuring a Shared File System.”</p> <p>Note – Do <i>not</i> use this keyword if your file system is in an Oracle Solaris Cluster environment.</p>
Disk partition or slice description	127 characters	<p>A /dev/ entry in this field identifies a disk partition or slice.</p>
Automated library or optical drive description	127 characters	<p>A /dev/samst entry identifies an automated library or optical drive. If you are configuring a network-attached automated library, see Chapter 4, “Creating Parameters Files for Network-Attached Automated Libraries,” in <i>Sun Storage Archive Manager 5.3 Configuration and Administration Guide</i>.</p>
Tape drive description	127 characters	<p>This entry can be in one of two forms:</p> <ul style="list-style-type: none"> ■ A /dev/rmt entry. ■ A path to a symbolic link that points to the same file to which the /dev/rmt link points. If you specify a tape drive in this manner, be sure to create the link before mounting the file system.

The Equipment Number Field (Required)

For each row in the mcf file, the equipment number (eq) field must contain a *unique* numeric identifier for the file system component or device being defined. This number must be an integer between 1 and 65534, inclusive.

Tip – Use low numbers to keep the internal software tables small.

The Equipment Type Field (Required)

The required Equipment Type field provides information that the software uses to determine how to interact with a particular device. Enter the two- or three-character mnemonic for the device type.

Some equipment can use the generic equipment types of od (optical disk), tp (tape), and rb (robot). For a file system, the following table describes specific Equipment Type codes:

TABLE 2-1 Equipment Type Field

Equipment Type	Description
ms	Defines a file system that stores both data and metadata on the same device (an md device).
ma	Defines a file system that stores metadata on a separate device (an mm device). The data in an ma file system can be stored on md, mr, or g_XXX devices.
md	Defines a striped or round-robin device that uses dual allocation for storing file data. See also “Dual and Single Allocation Schemes” on page 21.
mm	Defines a metadata device for storing inode and other metadata information. You can specify multiple metadata devices. Metadata (including inodes, directories, allocation maps, and so on) on ma file systems is located on metadata devices, separated from the file data devices. By default, metadata is allocated using round-robin allocation if you have multiple metadata devices.
mr	Defines a round-robin or striped data device that uses single allocation for storing file data. See also “Dual and Single Allocation Schemes” on page 21.
g XXX	Defines a striped data device. Striped groups start with the letter g followed by a number. The number must be an integer between 0 and 127, inclusive; for example, g12. All members in a striped group must be the same type and size. Different striped groups within one file system are not required to have the same number of members. md, mr, and g XXX devices cannot be mixed in one file system. Data can be striped (if all groups contain the same number of devices) or round-robin between groups. The default is round-robin.

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

Family Set Field (Required)

The family set field contains the name for a group of devices.

Family set names must start with an alphabetic character and can contain only alphabetic characters, numeric characters, or underscore (`_`) characters. The family set name cannot be longer than 31 characters.

The family set names are described in the following table.

Family Set	Description
File system name	All disk devices in the file system must use the same file system name in this field. The software uses the family set name to group devices together as a file system. It physically records the family set name on all of the devices in the file system when the <code>sammkfs</code> command is issued. You can change this name by issuing the <code>samfsck</code> command with the <code>-F</code> and <code>-R</code> options. For more information about the <code>sammkfs</code> command, see the <code>sammkfs(1M)</code> man page. For more information about the <code>samfsck</code> command, see the <code>samfsck(1M)</code> man page.
Automated library identifier	The library and all its associated drive devices must use the same identifier.
-	The dash character (<code>-</code>) indicates a stand-alone removable media device.

You can create a comment that is associated with a specific family set by inserting the identifier `#family-set-name:` just before the first device in that family set. Any comments that are added between that comment line and the last device in the family set will be associated with that family set. If the family set is later deleted through the SAM-QFS Manager software, any related comments will also be deleted from the `mcf` file.

Device State Field (Optional)

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

Additional Parameters Field (Optional)

For an automated library device, the Additional Parameters field is optional and can be left blank. By default, library catalog files are written to `/var/opt/SUNWsamfs/catalog/family-set-name`. Use this field if you want to specify an alternative path to the library catalog file.

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

For other entries, insert a dash (-) or leave this field blank.

Examples of mcf Files

Each file system configuration is unique. System requirements and actual hardware differ from site to site. The following code examples show sample mcf files. More complete examples that include information about how you can duplicate the configuration are available in [Chapter 3, “mcf File Examples.”](#)

EXAMPLE 2-1 mcf File Showing Striped Groups

This example shows an mcf file for a Sun QFS file system with two striped groups.

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

EXAMPLE 2-2 mcf File Showing Three File Systems

This example shows an mcf file with three file systems.

```
# SAM-QFS file system configuration example
#
# Equipment Eq Eq Fam. Dev. Additional
# Identifier Ord Type Set State Parameters
#-----
qfs1 10 ma qfs1 -
/dev/dsk/c1t13d0s6 11 mm qfs1 -
/dev/dsk/c1t12d0s6 12 mr qfs1 -
#
qfs2 20 ma qfs2 -
/dev/dsk/c1t5d0s6 21 mm qfs2 -
/dev/dsk/c5t1d0s6 22 mr qfs2 -
```

EXAMPLE 2-2 mcf File Showing Three File Systems (Continued)

```
#
qfs3 30 ma qfs3 -
/dev/dsk/c7t1d0s3 31 mm qfs3 -
/dev/dsk/c6t1d0s6 32 mr qfs3 -
/dev/dsk/c6t1d0s3 33 mr qfs3 -
/dev/dsk/c5t1d0s3 34 mr qfs3 -
```

EXAMPLE 2-3 mcf File Showing a File System and a Library

This example shows an mcf file with one archiving file system that uses md devices. This mcf file also defines a tape library.

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

Interactions Among File Settings, Options, and Directives Introduction

The mcf file defines each file system, but file system behavior depends on interactions among default system settings in the /etc/vfstab file, settings in the samfs .cmd file, and options in the mount command.

You can specify some mount options, such as the stripe width, in more than one place. Note that settings in one place can override the settings in another.

For information about the various ways to specify mount options, see [Chapter 8, “Setting Up Mount Parameters and Initializing the File System Environment,”](#) in *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide*.

mcf File Examples

The master configuration file, `/etc/opt/SUNWsamfs/mcf`, defines the topology of the equipment managed by the 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 file systems.

This chapter provides some specific examples of `mcf` files for various types of file systems.

Configuration Examples for Local File Systems

Use the configuration examples in this section for configuring the `mcf` file for a file system to be installed on a single Oracle Solaris OS host.

For examples that you can use in an Oracle Solaris Cluster environment, see [“Configuration Examples for Highly Available File Systems” on page 57](#).

Simple File System Configuration Example

This example shows how to configure two file systems using a server that has a SCSI-attached StorageTek Multipack desktop array from Oracle.

You can use the `format` command to determine how the disks are partitioned. [Example 3-1](#) shows the `format` command output.

Note – Only the last lines of `format` command output are shown.

EXAMPLE 3-1 `format` Command Output for Configuration Example

```
# format < /dev/null
Searching for disks...done
```

EXAMPLE 3-1 format Command Output for Configuration Example (Continued)

```

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/rdisk/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
  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

```

▼ How to Configure a Simple File System

Begin writing the `mcf` file for this configuration example by defining the file system and its disk partitions, as described in this procedure.

- 1 **Write the `mcf` file.**
- 2 **Add an `ma` entry for the first file system (`qfs1`).**
- 3 **Using the information from the output of the `format` command, add an `mm` entry listing the partitions that constitute the metadata for the `qfs1` file system.**
- 4 **Using the information from the output of the `format` command, add a series of `mr` entries listing the partitions that constitute the file data for the `qfs1` file system.**

5 Add 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).

The following code example shows the resulting `mcf` file.

```
# 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*
*/dev/dsk/c8t2d0s5 21 mm qfs2 on*
*/dev/dsk/c6t2d0s5 22 mr qfs2 on*
*/dev/dsk/c6t3d0s5 23 mr qfs2 on*
```

6 Modify the `/etc/vfstab` file.

Add 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 the code example below show entries for these new file systems.

For a description of the fields in the `/etc/vfstab` file, see Table 3–2.

```
# 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/rdisk/c0t10d0s0 /           ufs         1         no   logging
swap          -              /tmp        tmpfs       -         yes  -
*qfs1 - /qfs1 samfs - yes stripe=1*
*qfs2 - /qfs2 samfs - yes stripe=1*
```

Round-Robin Configuration Example

This example illustrates the configuration of a file system called `qfs3` 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).

▼ How to Configure the System for Round-Robin Allocation

This example introduces the round-robin data layout.

1 Write the `mcf` file as described in “Simple File System Configuration Example” on page 47.

The following code example shows the `mcf` file for this round-robin disk configuration.

```
# 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. The following code example shows `stripe=0` for the `qfs3` file system.

```
# cat /etc/vfstab
#device          device          file          mount
#to              to              system        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/rdisk/c0t10d0s0 /             ufs         1          no         logging
swap             -               /tmp          tmpfs       -          yes        -
*qfs3 - /qfs3 samfs - yes stripe=0*
```

3 Initialize the file system by using the `sammkfs` command.

The default disk allocation unit (DAU) is 64 kilobytes, but the following example sets the DAU size to 128 kilobytes.

```
# sammkfs -a 128 qfs3
```

Local Striping Configuration Example

This example illustrates the configuration of a file system called `qfs4` 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, logical unit number (LUN) 0.
- The data devices consist of four disks attached to controller 6. Each disk is on a separate target (2—5).

▼ How to Configure the System for Local Striping

1 Write the `mcf` file as shown in “Simple File System Configuration Example” on page 47.

The following code example shows the `mcf` file for this striped disk configuration.

```
# 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. The following code example shows the `/etc/vfstab` file with a mount parameter of `stripe=1` set for the `qfs4` file system.

```
# cat /etc/vfstab
#
#device          device          file            mount
#to             to             point          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/rdisk/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` command to initialize the file system.

3 Initialize the StorageTek QFS file system by using the `sammkfs` command.

The following example sets the DAU size to 128 kilobytes.

```
# sammkfs -a 128 qfs4
```

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.

Striped Group Configuration Example

Striped groups enable you to build RAID-0 devices of separate disk devices. Striped groups, however, have 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. You cannot 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 file system (called `qfs5`) that separates the metadata onto a low-latency disk. The `mc f` 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).

▼ How to Configure the System for Striped Groups

1 Write the `mc f` file as shown in “Simple File System Configuration Example” on page 47.

The following code example shows a sample `mc f` file for a striped group configuration.

```
# cat /etc/opt/SUNWsamfs/mc f
#
# 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. The following code example 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`.

```
# cat /etc/vfstab
#device          device          mount      file          mount
#to              to              point      system        at          mount
#mount          fck             pass       type          boot       params
#-----
fd               -               /dev/fd    fd            -          no         -
/proc           -               /proc      proc          -          no         -
/dev/dsk/c0t10d0s1 -               -          swap          -          no         -
/dev/dsk/c0t10d0s0 /dev/rdisk/c0t10d0s0 /          ufs           1          no         logging
swap            -               /tmp       tmpfs         -          yes        -
*qfs5 - /qfs5 samfs - yes stripe=0*
```

3 Initialize the file system by using the `sammkfs` 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.

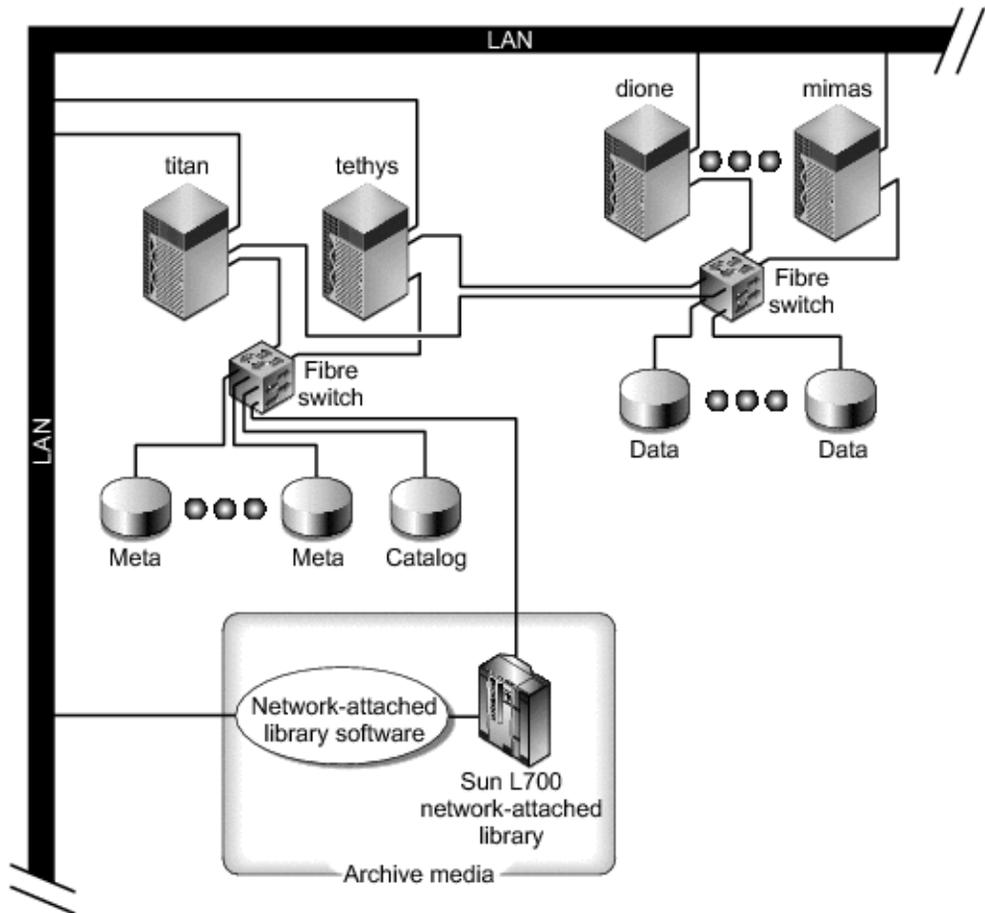
```
# sammkfs qfs5
```

This example has 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-robin 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` command.

Configuration Example for a Shared File System on an Oracle Solaris OS Platform

The following figure illustrates a shared file system configuration in an archiving environment.

FIGURE 3-1 Shared File System Configuration



This figure 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 namespace, 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 StorageTek QFS shared file system. The data disks are connected to all four hosts by a Fibre Channel (FC) connection.

▼ How to Configure the Shared File System

1 Issue the `format` command and examine its output.

Make sure that the metadata disk partitions configured for the shared file system mount point are connected to the potential metadata servers. Also make sure that the data disk partitions configured for the 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` command might display multiple controllers. These correspond to the multiple paths to the actual devices.

The following code example shows the `format` command output for `titan`. There is one metadata disk on controller 2, and three data disks on controller 3.

```
# 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>
   /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
```

The following code example shows the `format` command output for `tethys`. There is one metadata disk on controller 2, and four data disks on controller 7.

```
# 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 this code example:

- 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`, which is the same metadata disk as `tethys`'s controller 2, target 0.

The following code example shows the `format` command output for `mimas`. There are three data disks on controller 1 and no metadata disks.

```

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 the code examples, 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.

2 Create the `mc f` file on the metadata server.

The only difference between the `mc f` file of a shared file system and an unshared file system is the presence of the `shared` keyword in the Additional Parameters field of the file system name line for a shared file system.

Note – If file systems are already operational on the shared file system's metadata server or on any of the client host systems, select a family set name and select equipment numbers that do not conflict with existing family set names or equipment numbers on any host that will be included in the shared file system.

The following code example shows a fragment of the `mcf` file for `titan` that defines several disks for use in the shared file system. It shows the `shared` keyword in the Additional Parameters field on the file system name line.

```
# Equipment          Eq  Eq  Family  Dev  Addl
# Identifier         Ord Type Set   Stat Params
#-----
sharefs1            10  ma  sharefs1  on  shared
/dev/dsk/c2t50020F23000065EE0s6 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 Oracle Solaris Cluster software moves a highly available file system from a failing node to a viable node in the event of a node failure.

Each node in the Oracle Solaris 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 Oracle Solaris Cluster environment.

▼ How to Create an `mcf` File for a Highly Available File System

- 1 Add an `ma` entry for the file system.
- 2 Add an `mm` entry listing the partitions that constitute the metadata for the `qfs1` file system.
- 3 Add 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 `sccidadm` command to determine the partitions to use.

The following code example shows an `mcf` file entry for a highly available file system that uses 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
```

The following code example shows an `mcf` file entry for a highly available file system that uses Solaris Volume Manager metadevices. In this example, the Solaris Volume Manager metaset in use is named `red`.

```
#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
```

The following code example shows an `mcf` file entry for a highly available file system that uses 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 an Oracle Solaris Cluster Platform

In this example, `ash` and `elm` are nodes in an Oracle Solaris 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. .

▼ How to Create an `mcf` File for a Shared File System in an Oracle Solaris Cluster Environment

You must create the `mcf` file on the node that you want to designate as the metadata server.

- 1 Use the `scdidadm -L` command to obtain information about the devices included in the Oracle Solaris Cluster environment.

The `scdidadm` command administers the device identifier (DID) devices. The `-L` option lists all the DID device paths, including those on all nodes in the Oracle Solaris Cluster environment.

The following code example uses StorageTek 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.

```
ash# scdidadm -L
1      ash:/dev/rdisk/c0t6d0      /dev/did/rdisk/d1
2      ash:/dev/rdisk/c1t1d0      /dev/did/rdisk/d2
3      ash:/dev/rdisk/c1t0d0      /dev/did/rdisk/d3
```

```

4      elm:/dev/rdisk/c6t50020F2300004921d1 /dev/did/rdsk/d4
4      ash:/dev/rdisk/c5t50020F2300004921d1 /dev/did/rdsk/d4
5      elm:/dev/rdisk/c6t50020F2300004921d0 /dev/did/rdsk/d5
5      ash:/dev/rdisk/c5t50020F2300004921d0 /dev/did/rdsk/d5
6      elm:/dev/rdisk/c6t50020F23000049CBd1 /dev/did/rdsk/d6
6      ash:/dev/rdisk/c5t50020F23000049CBd1 /dev/did/rdsk/d6
7      elm:/dev/rdisk/c6t50020F23000049CBd0 /dev/did/rdsk/d7
7      ash:/dev/rdisk/c5t50020F23000049CBd0 /dev/did/rdsk/d7
8      elm:/dev/rdisk/c6t50020F23000055A8d0 /dev/did/rdsk/d8
8      ash:/dev/rdisk/c5t50020F23000055A8d0 /dev/did/rdsk/d8
9      elm:/dev/rdisk/c6t50020F23000078F1d0 /dev/did/rdsk/d9
9      ash:/dev/rdisk/c5t50020F23000078F1d0 /dev/did/rdsk/d9
10     elm:/dev/rdisk/c0t6d0           /dev/did/rdsk/d10
11     elm:/dev/rdisk/c1t1d0          /dev/did/rdsk/d11
12     elm:/dev/rdisk/c1t0d0          /dev/did/rdsk/d12

```

2 Using the output from the `scdidadm -L` command, use the `format` command to display the information for the devices in the Oracle Solaris Cluster environment.

The following code example shows the `format` command output from all the `/dev/did` devices. You will need this information when you build the `mcf` file.

```
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
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/rdisk/d6s2
selecting /dev/did/rdisk/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/rdisk/d7s2
selecting /dev/did/rdisk/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

```

ash# format /dev/did/rdisk/d8s2
selecting /dev/did/rdisk/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/rdisk/d9s2
selecting /dev/did/rdisk/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` command displays the space available on a device, but it does not indicate whether a disk is mirrored or striped. This code example's `format` command output reveals the following information that is used during creation of the `mcf` file shown in the next code example:

- Output for devices `d4s0` and `d6s0` shows 16.86 Gbytes each. These devices are assigned equipment numbers 501 and 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 number 503 and 504, respectively, in the `mcf` file. They are the appropriate size to be used for storing data.

3 Add an `ma` entry for the file system.

Include the `shared` keyword in the Additional Parameters field.

4 Add 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 Add a series of `mr` entries listing the partitions that constitute the file data for the `qfs1` file system.

The following code example shows the resulting `mcf` file.

```
#Equipment      Eq  Eq   Family  Additional
#Identifier      Ord Type Set      Parameters
#-----
#
# Family Set sqfs1 (shared FS for SunCluster)
#
sqfs1            500  ma   sqfs1   shared
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   -
```

Configuring the File System

This chapter provides information about configuring an archiving or stand-alone file system. For information about configuring a shared file system, see [Chapter 5, “Configuring a Shared File System.”](#)

Function of the mcf File

The master configuration file (`mcf`), located in `/etc/opt/SUNWsamfs/mcf`, describes all devices that the Sun QFS or SAM-QFS software controls or uses. When you create this ASCII file at system configuration time, you declare attributes for each device and group the devices in each file system into family sets.

The `mcf` file contains the information that the file systems need in order to identify and organize RAID and disk devices into file systems. It also contains entries for each automated library or device included in a file system. A sample `mcf` file is located in `/opt/SUNWsamfs/examples/mcf`.

For more information about the `mcf` file, see [Chapter 2, “About the Master Configuration File,”](#) and the `mcf(4)` man page.

Initializing a File System

To create a new file system or to replace an old or damaged file system, use the `sammkfs (1M)` command to initialize the file system. The `sammkfs` command constructs and initializes file systems.

Note – Beginning with Sun QFS 5.0, the `sammkfs` command creates a version 2A file system that has new features but is not compatible with previous releases. Use the `sammkfs -P` format to create a version 2 file system that is backwards compatible with previous releases. Use the `sammkfs -a allocation-unit` option to specify the DAU setting.

The following `samfsinfo` command output shows that the `samfs1` file system is using a version 2 superblock.

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

The following example shows the `sammkfs` command in its simplest form, with the file system name as its only argument. This command builds a version 2A superblock for a stand-alone Sun QFS or SAM-QFS file system.

```
# sammkfs samqfs1
```

For more information about the `sammkfs` command, its options, and the implications of the version 2 and 2A superblocks, see the `sammkfs(1M)` man page. For information about using the `sammkfs` command to initialize a shared Sun QFS file system, see [Chapter 5, “Configuring a Shared File System.”](#)

Configuration Examples

This section presents sample configurations and shows various steps and decisions involved in setting up the `mcf` file on a server.

Note that all sample configurations could have automated libraries and other removable media devices defined as well, essentially extending the file system beyond the size of the disk cache. Removable media device configurations are shown in only one example. For information about configuring removable media devices, see [Chapter 2, “Configuring Storage Devices for Archiving,”](#) in *Sun Storage Archive Manager 5.3 Configuration and Administration Guide*.

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

▼ How to Create a Round-Robin Disk Configuration

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

This procedure assumes the following:

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

1 Create the `mcf` file, as shown in the following code example.

```
# Sun QFS disk cache configuration
# Round-robin mcf example
# Equipment      Eq   Eq   Fam.  Dev   Additional
# Identifier      Ord  Type Set   State Parameters
#-----
qfs1              1   ma   qfs1
/dev/dsk/c5t0d0s6 11  mm   qfs1  on
/dev/dsk/c1t1d0s6 12  mr   qfs1  on
/dev/dsk/c2t1d0s6 13  mr   qfs1  on
/dev/dsk/c3t1d0s6 14  mr   qfs1  on
/dev/dsk/c4t1d0s6 15  mr   qfs1  on
```

2 Create the `/qfs` mount point for the `/qfs1` file system.

```
# mkdir /qfs
```

3 Use the `sammkfs` command to initialize the file system.

The following example uses the default 64-kilobyte DAU.

```
# sammkfs qfs1
```

4 Modify the `/etc/vfstab` file.

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

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

5 Mount the file system.

```
# mount /qfs
```

▼ How to Create a Striped Disk Configuration

In this sample configuration, file data is striped to four data partitions by default.

This procedure assumes the following:

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

1 Create the mcf file, as shown in the following code example.

```
# Sun QFS disk cache configuration
# Striped Disk mcf example
# Equipment      Eq   Eq   Fam.  Dev.  Additional
# Identifier     Ord  Type Set   State Parameters
#-----
qfs1              10   ma   qfs1
/dev/dsk/c0t1d0s6 11   mm   qfs1   on
/dev/dsk/c1t1d0s6 12   mr   qfs1   on
/dev/dsk/c2t1d0s6 13   mr   qfs1   on
/dev/dsk/c3t1d0s6 14   mr   qfs1   on
/dev/dsk/c4t1d0s6 15   mr   qfs1   on
```

2 Create the /qfs mount point for the /qfs1 file system.

```
# mkdir /qfs
```

3 Use the sammkfs command to initialize the file system.

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

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

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

4 Modify the /etc/vfstab file.

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

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

5 Mount the file system.

```
# mount /qfs
```

▼ How to Create a Striped Group Configuration

Striped groups enable you to group RAID devices together for very large files. A DAU is represented by one bit in the bitmap. If the striped group has n devices, n multiplied by the DAU is the minimum allocation. Only one bit in the bitmap is used to represent $n \times$ DAU.

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

$$\text{minimum-disk-space-allocated} = \text{DAU} \times \text{number-of-disks-in-the-group}$$

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

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

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

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

This procedure assumes the following:

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

1 Create the `mcf` file, as shown in the following code example.

```
# Sun QFS disk cache configuration
# Striped Groups mcf example
# Equipment      Eq  Eq   Fam.  Dev.  Additional
# Identifier     Ord Type Set   State Parameters
#-----
qfs1              10  ma  qfs1
/dev/dsk/c0t1d0s6 11  mm  qfs1   on
/dev/dsk/c1t1d0s6 12  g0  qfs1   on
/dev/dsk/c2t1d0s6 13  g0  qfs1   on
/dev/dsk/c3t1d0s6 14  g1  qfs1   on
/dev/dsk/c4t1d0s6 15  g1  qfs1   on
```

2 Create the `/qfs` mount point for the `/qfs1` file system.

```
# mkdir /qfs
```

3 Use the `sammkfs` command to initialize the file system.

The following example sets the DAU size to 128 kilobytes.

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

4 Modify the `/etc/vfstab` file.

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

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

This `/etc/vfstab` file sets the stripe width using the `stripe=` option. This example has two striped groups, `g0` and `g1`. With the `stripe=0` specification, files are written round-robin around the two striped groups.

Note – To change the configuration of the striped group after it is created, you must issue another `sammkfs` command.

5 Mount the file system.

```
# mount /qfs
```

Configuring a Shared File System

This chapter provides information about configuring a shared Sun QFS file system.

For information about configuring a shared file system in an Oracle Solaris Cluster environment, see [Chapter 4, “Configuring Sun QFS Shared File Systems With Oracle Solaris Cluster,”](#) in *Using Sun QFS and Sun Storage Archive Manager with Oracle Solaris Cluster*.

Using Shared QFS With NFS

Note – If you are using NFS v4, you must disable delegations before you can use shared QFS.

Starting with SAM-QFS 5.0 on Oracle Solaris 10, the Service Management Facility (SMF) is used to manage the mounting of the file system at boot time. If your file system uses NFS, the exact sequence in which you configure NFS and shared QFS is important. If you do not follow the steps in the following procedure, either the shared QFS mount or the NFS share will succeed and the other will fail.

▼ How to Configure Shared Sun QFS With NFS

- 1 Export the existing NFS configuration to a file.

The following example exports the configuration into a file `/var/tmp/server.xml`.

```
# svccfg export /network/nfs/server > /var/tmp/server.xml
```

- 2 In the exported file, after the local file system dependency, add a dependency to mount QFS file systems before you NFS share them.**

For example:

```
<!--  
  Must have QFS filesystems mounted before sharing them  
-->  
  <dependency name='qfs'  
    grouping='require_all'  
    restart_on='error'  
    type='service'>  
    <service_fmri value='svc:/network/qfs/shared-mount:default' />  
  </dependency>
```

- 3 Validate the changes that you made to the file.**

```
# svccfg validate /var/tmp/server.xml
```

- 4 Disable NFS.**

```
# svcadm disable nfs/server
```

- 5 Delete the existing NFS server configuration.**

```
# svccfg delete nfs/server
```

- 6 Import the file that you edited into SMF.**

```
# svccfg import /var/tmp/server.xml
```

- 7 Enable NFS.**

NFS uses the updated file and reads the Sun QFS dependency information.

```
# svcadm enable nfs/server
```

- 8 Confirm that the dependency is applied.**

```
# svcs -d svc:/network/nfs/server:default
```

Mounting and Unmounting Shared File Systems

When you mount or unmount a shared file system, the order in which you mount or unmount the metadata server and the clients is important.

For failover purposes, the mount options should be the same on the metadata server and all potential metadata servers. For example, you can create a `samfs.cmd` file that contains the mount options and copy that file to all of the hosts.

For more information about mounting shared file systems, see [Chapter 12, “Mount Options in a Shared File System,”](#) and the `mount_samfs(1M)` man page.

▼ How to Mount a Shared File System

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

- 2 **Mount the metadata server.**

Mount the file system on the metadata server before you mount the file system on any client hosts. For example:

```
# mount -F samfs qfs1 /qfs -o shared
```

Tip – If the mount information has been placed in `/etc/vfstab`, you can use the simpler command:

```
# mount qfs1
```

For information about the `/etc/vfstab` file and other mount options, see [Chapter 8, “Setting Up Mount Parameters and Initializing the File System Environment,”](#) in *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide*.

- 3 **Mount the client hosts.**

You can mount the file system on the client hosts in any order. For more information, see the `mount(1M)` man page.

▼ How to Unmount a Shared File System

Note – If the file system is shared through NFS or SAMBA, unshare the file system before you unmount it.

- 1 **Unmount the file system on every participating client host.**

For example:

```
client# umount /samqfs
```

If necessary, use the `-f` option to the `umount` command to force a file system to unmount.

Note – Forced unmount of a shared client might not complete if the file system is not mounted on the metadata server.

For more information on unmounting procedures, see [“Unmounting File Systems”](#) in *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide* and the `umount(1M)` man page.

You can also use the `-o await_clients#` flag with the `umount` command to cause the unmount process to wait a specified number of seconds (`#`) for clients to unmount. At the end of the

waiting period, or as soon as all clients have unmounted, the unmount proceeds. If this argument is specified for a non-shared file system, or if the host is not the metadata server for the shared file system, the option will be ignored.

This flag can also be used in conjunction with the `-f` option. In this case, the software will wait for the specified time period before forcing the unmount.

2 Unmount the file system on the metadata server.

```
metaserver# umount /samqfs
```

At unmounting time, several conditions can be present in a file system that might require you to issue the `umount` command a second time.

Note – If the file system still does not unmount, use `unshare`, `fuser`, or another command in conjunction with the `umount` command.

See Also For more information on unmounting procedures, see “[Unmounting File Systems](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide* and the `umount(1M)` man page.

Adding or Removing a Client Host

This section provides instructions for adding and removing client host systems in a shared file system.

▼ How to Add a Client Host to a Shared File System

You can add a client host to a shared file system after you have configured and mounted the file system on all participants.

- 1 **Become superuser on the metadata server.**
- 2 **Use the `samsharefs` command to retrieve the current shared file system information and write it to an editable file.**

- If the shared file system is mounted, issue the `samsharefs` command on the current metadata server. For example:

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

- If the shared file system is not mounted, issue the `samsharefs` command with the `-R` option from the metadata server or from any of the potential metadata servers. For example:

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

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

3 Open the shared file system information file.

For example:

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

4 Add a line for the new client host.

The following code example shows the file after addition of the line for a host named `helene` as the last line.

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

5 Use the `samsharefs` command to update the current information in the binary file.

The options to use with this command, and the system from which this command is issued, depend on whether the Sun QFS shared file system is mounted, as follows:

- If the file system is mounted, issue the `samsharefs -u` command from the current metadata server. For example:

```
# samsharefs -u sharefs1
```

- If the file system is not mounted, issue the `samsharefs -R -u` command from the active metadata server or from any of the potential metadata servers. For example:

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

The client host `helene` is now recognized.

6 As superuser, log in to the client host to be added.

7 Use the `format` command to verify the presence of client host disks.

8 Update the mcf file on the client host.

Before a host system can access or mount a shared file system, it must have that file system defined in its mcf file. The mcf file must be updated to match all client hosts in the shared file system. The file system and disk declaration information must have the same data for the Family Set Name, Equipment Number, 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, since controller assignments will probably differ from host to host.

For information, see [“Updating the mcf file in a Shared File System Environment” on page 77](#).

9 Notify the sam-fsd daemon of the configuration changes by issuing the samd config command on the metadata server host.

```
# samd config
```

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

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

For information on creating the local hosts file, see [“Creating the Local Hosts Configuration File” on page 79](#).

If you create this file, use the samd config command on the client host to notify the sam-fsd daemon of the configuration changes.

```
# samd config
```

11 Verify that the sam-sharefsd daemon is running for this file system.

Use the ps and grep commands as shown in the following code 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
```

The code example above shows that the sam-sharefsd daemon is active for the sharefs1 file system.

12 If the new Sun QFS shared file system does not already have a mount point, create the directory for the mount point.

For example:

```
# mkdir /sharefs1
```

13 Give the mount point the 755 set of permissions.

For example:

```
# chmod 755 /sharefs1
```

The permissions must be the same on all participant hosts. 755 is suggested as the initial permission set because users must have execute permission on the mount point in order to be able to use the file system after it has been mounted. After you mount the file systems, the root directory's permissions override this setting.

14 Modify the `/etc/vfstab` file.

You must have an entry in the `/etc/vfstab` file for the Sun QFS shared file system. Specify shared in the Mount Parameters field. In addition, do one of the following:

- **If you do not want to mount this file system automatically at boot time, type no in the `Mt@boot` field.**
- **If you do want the Sun QFS shared file system to automatically mount at boot, do the following:**

a. **Type yes in the `Mt@boot` field.**

b. **Add the `bg` mount option in the `Mt params` field.**

The `bg` mount option mounts the file system in the background if the metadata server is not responding.

The following code example shows the shared and `bg` entries in the `Mt params` field.

```
# File /etc/vfstab
# FS name  FS to fsck  Mnt pt   FS type  fsck  Mt@boot  Mt params
#
sharefs1  -           /sharefs1 samfs    -      yes      shared,bg
```

15 Verify that the file system is mounted on the metadata server.

```
# df -k
```

The file system should be included in the displayed list.

16 From the client host, mount the Sun QFS shared file system.

For example:

```
# mount /sharefs1
```

For more information about mounting Sun QFS shared file systems, see [Chapter 12, “Mount Options in a Shared File System,”](#) or see the `mount_samfs(1M)` man page.

▼ How to Remove a Client Host From a Shared File System

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

Note – You can use the `samsharefs` command to verify that you are logged in to the metadata server or a client host.

2 Unmount the shared file system on each client host on which the shared file system is mounted.

For example:

```
client# umount sharefs1
```

3 Unmount the shared file system on the metadata server.

For example:

```
metaserver# umount sharefs1
```

4 Use the `samsharefs` command to obtain the current configuration information.

The following example command writes current configuration information to file `/etc/opt/SUNWsamfs/hosts.sharefs1`.

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

5 Open the shared file system information file.

The following code example shows the file before the client host is deleted.

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

6 In the file, delete the client host or hosts that are no longer to be supported.

The following code example shows the file after the line for `helene` has been deleted.

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

7 Use the `samsharefs -R -u` command to update the current hosts information.

For example:

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

The host `helene` has been removed.

8 Use the `samsharefs -R` command to display the current configuration.

For example:

```
# samsharefs -R sharefs1
```

9 Mount the shared file system, first on the metadata server and then on each client host in the file system.

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

Updating the `mcf` file in a Shared File System Environment

The `samfsconfig` command generates configuration information that can help you to identify the devices included in the shared file system. You can then use this information to update the `mcf` files on each client host.

Issue a separate `samfsconfig` 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.

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

EXAMPLE 5-1 `samfsconfig` Command Example on `tethys`

The following example shows the `samfsconfig` command being 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`, another metadata server in the shared file system.

```
tethys# samfsconfig /dev/dsk/*
#
# Family Set &rdquo;sharefs1&rsquo; Created Wed Jun 27 19:33:50 2003
#
sharefs1                10 ma sharefs1 on shared
/dev/dsk/c2t50020F23000065EE0s6 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
```

Copy the last five lines of output from the `samfsconfig` command into the `mcf` file on client host `tethys`. Verify the following:

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

The next example shows the resulting `mcf` file.

EXAMPLE 5-2 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

```

EXAMPLE 5-3 samfsconfig Command Example on mimas

The following example shows the `samfsconfig` command being used to retrieve device information for family set `sharefs1` on client host `mimas`. In this example, `mimas` can never become a metadata server, and it is not connected to the metadata disks.

```

mimas# samfsconfig /dev/dsk/*
#
# Family Set "sharefs1" Created Wed Jun 27 19:33:50 2001
#
# Missing slices
# Ordinal 0
# /dev/dsk/clt50020F2300005D22d0s6 12 mr sharefs1 on
# /dev/dsk/clt50020F2300006099d0s6 13 mr sharefs1 on
# /dev/dsk/clt50020F230000651Cd0s6 14 mr sharefs1 on

```

In the output from the `samfsconfig` command on `mimas`, note that Ordinal 0, which is the metadata disk, is not present. For devices that are missing, the `samfsconfig` process comments out the elements of the file system and omits the file system family set declaration line. Make the following edits to the `mcf` file:

- Create a file system family set declaration line, beginning with `sharefs1`, in the `mcf` file for client host `mimas`. Put 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 number entry. For each of these lines, the keyword `nodev` must appear in the Equipment Identifier field for the inaccessible device.
- Ensure that each Device State field is set to `on`.
- Uncomment the device lines.

The following example shows the resulting `mcf` file for `mimas`.

EXAMPLE 5-4 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

```

EXAMPLE 5-4 mcf File for Client Host mimas (Continued)

```
/dev/dsk/c1t50020F2300006099d0s6 13 mr sharefs1 on
/dev/dsk/c1t50020F230000651Cd0s6 14 mr sharefs1 on
```

Creating the Local Hosts Configuration File

The local hosts configuration file must reside in the following location:

```
/etc/opt/SUNWsamfs/hosts._family-set-name_.local
```

Comment lines must begin with a hash character (#). Characters to the right of the hash character are ignored.

The following table shows the fields in the local hosts configuration file.

TABLE 5-1 Local Hosts Configuration File Fields

Field	Content
Host Name	This field must contain the alphanumeric name of a metadata server or potential metadata server that is part of the Sun QFS shared file system.
Host Interfaces	<p>This field must contain a comma-separated list of host interface addresses. Use the output from the <code>ifconfig -a</code> command. The individual interfaces can be specified in one of the following ways:</p> <ul style="list-style-type: none"> ▪ Dotted-decimal IP address form ▪ IP version 6 hexadecimal address form ▪ As a symbolic name that the local domain name service (DNS) can resolve to a particular host interface <p>Each host uses this field to determine whether it will try to connect to the specified host interface. The system evaluates the addresses from left to right, and the connection is made using the first responding address in the list that is also included in the shared hosts file.</p>

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

Note – “Client” as in “network client”, is used to refer to both client hosts and the metadata server host.

The metadata server and the client hosts use both the `/etc/opt/SUNWsamfs/hosts.fsname` file on the metadata server and the `hosts.fsname.local` file on each client host (if it exists) to determine the host interface to use when accessing the file system. This process is as follows:

1. The client obtains the list of metadata server host IP interfaces from the file system's on-disk host file.

To examine this file, issue the `samsharefs` command from the metadata server or from a potential metadata server.

2. The client searches its files for a `hosts.fsname.local` file.

Depending on the outcome of the search, one of the following courses of action is taken:

- If a `hosts.fsname.local` file does not exist, the client attempts to connect, in turn, to each address in the system hosts configuration file until it succeeds.
- If the `hosts.fsname.local` file exists, the client performs the following actions:
 1. Compares the list of addresses for the metadata server from both the `/etc/opt/SUNWsamfs/hosts.fsname` file on the metadata server and the `hosts.fsname.local` file.
 2. Builds a list of addresses that are present in both places, and then attempts to connect to each of these addresses, in turn, until it succeeds. If the order of the addresses differs in these files, the client uses the ordering in the `hosts.fsname.local` file.

The following example shows a detailed scenario for a shared file system that comprises four hosts.

EXAMPLE 5-5 Sun QFS Shared File System Hosts File Example

The following example shows a hosts file that lists four hosts.

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

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.

The following example shows the information in the `hosts.sharefs1.local` files on `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
```

EXAMPLE 5-5 Sun QFS Shared File System Hosts File Example (Continued)

Systems `mimas` and `dione` are not on the private network. To guarantee that they always connect to `titan` and `tethys` through `titan`'s and `tethys`'s public interfaces, the system administrator has created identical copies of `/etc/opt/SUNWsamfs/hosts.sharefs1.local` on `mimas` and `dione`.

The following example shows the information in the `hosts.sharefs1.local` files on `mimas` and `dione`.

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

Changing the Metadata Server

Changing the Metadata Server in a Shared File System Environment

The procedures in this section describe how to change the host that is acting as the metadata server in a shared file system without using the automatic Membership Services feature of a software package.

You can change the metadata server system manually under the following circumstances:

- If the metadata server becomes unavailable
- If you want to change the metadata server or the potential metadata servers

For a metadata server change to succeed, the mount options of the existing metadata server and all potential metadata servers must be the same.

If the metadata server of a shared file system crashes, it is safe to change the metadata server only after rebooting the metadata server or otherwise ensuring that the server cannot issue any I/O before being rebooted. *Do not use* any of the following methods to stop the server, because these are likely to corrupt the file system:

- Issuing an L1-A key sequence
- Performing an involuntary failover to another host
- Issuing a `go (continue)` command, requesting a dump file, or issuing a `sync` command to the old metadata server

Similarly, if the metadata server panics and drops into kernel adb command, do not change the metadata server and then issue a :c (continue) command on the server. This action causes the old metadata server to push stale buffers out to the now-active file system.

▼ How to Change the Metadata Server When the Metadata Server Is Available

- On the existing metadata server, issue the `samsharefs -s` command to declare the new metadata server.

For example:

```
titan# samsharefs -s tethys sharefs1
```

Note – In archiving environments, you should stop all archiving operations on the metadata server before you issue this command.

▼ How to Change the Metadata Server When the Metadata Server Is Not Available

- 1 Ensure that the existing metadata server cannot restart without being rebooted.

Specifically, ensure that the server is powered down, rebooted, halted, or disconnected from the metadata disks. Your goal is to bring down the old metadata server and flush or destroy all buffers or otherwise ensure that they cannot be rewritten.

Use the following key sequence from the `kadb` prompt:

```
kadb[1]: sync      # Forces a dump
kadb[1]: $q       # Exits the debugger for prom
```

Use the following key sequence from the PROM prompt:

```
{0} > sync          # Forces the buffers out
{0} > boot _args_   # Discards buffers
```

For *args*, specify arguments for the `boot` command, such as the `-r` or the `-v` option. For more information, see the `boot(1M)` man page.

- 2 From the new (potential) metadata server, wait at least the period of the maximum lease time, and then issue the `samsharefs` command.

For example:

```
# samsharefs -R -s tethys sharefs1
```

The wait ensures that all client leases expire before you issue the `samsharefs` command. If you are uncertain as to whether the lease time has expired, bring up the `samu(1M) N` display. For information about the `samu` command, see [Chapter 13, “Using the samu Operator Utility.”](#) For

information about leases and their durations, see “Using Leases in a Sun QFS Shared File System: (`rdlease`, `wrlease`, and `aplease` Options)” on page 166.



Caution – If you use the `-R` option to the `samshare fs` command on a mounted file system to change the metadata server host, you must first stop, disable, and disconnect the active metadata server. Failure to do so can cause file system corruption.

3 (Optional) Unmount the file system.

Perform this step only if you want to perform a file system check.

Use the procedure in “Unmounting File Systems” in *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide*.

4 (Optional) Issue the `samfsck` command to perform a file system check.

If the metadata server of a Sun QFS shared file system crashes, the server should be rebooted and the file system should be unmounted on all clients before the `samfsck` command is run. The server and clients preallocate blocks before changing the length of files. The `samfsck` command cleans up files that have extra blocks allocated, and these extra blocks might contain data. If this type of cleaned-up file is awaiting a size update from the client, the file will be missing those blocks when the client continues. As a result, the file will be missing data, and the missed data will read as zeroes.

Changing the Metadata Server in an Archiving Environment

The procedures in this section describe how to change the host that is acting as the metadata server in an archiving shared file system without using the automatic Membership Services feature of a software package.

You can change the metadata server system manually under the following circumstances:

- If the metadata server becomes unavailable
- If you want to change the metadata server or the potential metadata servers

For a metadata server change to succeed, the mount options of the existing metadata server and all potential metadata servers must be the same.

▼ How to Change the Metadata Server in an Archiving Environment

Archiving functions can run only on one host at any time. This procedure assumes that both systems are running at the time of the transfer. This example moves archiving functions from host A to host B.

Before carrying out this procedure, verify that host B has access to the robot catalog from host A. The `archiver.cmd` file, `mcf` file, `stager.cmd` file, and other configuration files must be identical to those on host A.

1 Idle archiving processes on host A.

a. Run the `samcmd aridle` and `samcmd stidle` commands to halt archiving and staging on host A.

These commands will allow current archiving and staging to complete but will not start any new work.

b. Idle all of the tape drives on host A.

Use `samcmd eq idle`, where `eq` is the equipment number of the drive. This command will put the drives in an off state after any current I/O completes.

c. When the archiver and stager are idle and the tape drives are all in the off state, run the `samd stop` command to halt all of the robot and tape-related daemons.

d. If you have a cron job that runs the recycler, remove this entry from the crontab and verify that the recycler is not currently running.

At this point, the archiving processes have been halted and file system failover to host B can be performed.

2 Start the archiving processes on host B by running the `samd config` command on host B.

This command causes `sam-fsd` and its subprocesses (archiver, stager, and so on) to reconfigure and re-read the configuration files. It also causes `sam-amld` and the tape-library-related daemons to start. At this point all Sun QFS shared client applications waiting for stages must reissue the stage requests.

Host B should now be fully functioning as the archiving process server and metadata server for all file systems.

Converting an Unshared File System to a Shared File System

To perform initial installation and configuration for a shared file system, follow the instructions in [Chapter 5, “Installing Sun QFS and SAM-QFS,” in *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide*](#). Many examples in this section use host names and configuration information that were introduced in that document.

Converting an unshared file system to a shared file system consists of two tasks:

- Converting the metadata server.
- Adding each client to the metadata server. This section describes these procedures.

▼ How to Convert an Unshared Metadata Server to a Shared Metadata Server

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

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

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

Depending on your software, these files might include the `mcf`, `archiver.cmd`, `defaults.conf`, `samfs.cmd`, or `inquiry.conf` files. Back up these files for all file systems. Also make sure that you have backup copies of files in the `/etc/opt/SUNWsamfs` directory and in the `/var/opt/SUNWsamfs` directory.

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

File systems should be backed up regularly according to your site's policies. If you are comfortable with the backup files that already exist for your file systems, you do not need to back them up again now.

4 Unmount the file system.

For instructions, see “[Unmounting File Systems](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide*.

5 Convert the file system to a Sun QFS shared file system using the `samfsck -S -F family-set-name` command.

For *family-set-name*, specify the family set name of the file system that you are converting to a new shared file system. For example:

```
# samfsck -S -F sharefs1
```

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

For example:

```
# Equipment Eq Eq Family Dev Add
# Identifier Ord Type Set State Params
# -----
sharefs1 10 ma sharefs1 on shared
/dev/dsk/c2t50020F23000065EE0s6 11 mm sharefs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12 mr sharefs1 on
/dev/dsk/c7t50020F2300006099d0s6 13 mr sharefs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14 mr sharefs1 on
```

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

For example:

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

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

For example:

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host      Host      IP          Server      Not Server
# Name      Addresses          Priority     Used Host
# ----      -
titan      titan-ge          0 1 - server
tethys     tethys-ge        0 2 - server
```

See “Creating the Shared Hosts File on the Metadata Server” in *Using Sun QFS and Sun Storage Archive Manager with Oracle Solaris Cluster* for more information about creating the hosts configuration file.

9 Run the `samsharefs -u -R family-set-name` command to initialize the file system and the host configuration.

For example:

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

Note – If you see an error message from this command, you can probably ignore the message.

10 Notify the `sam-fsd` daemon of the configuration changes by issuing the `samd config` command.

```
# samd config
```

11 Mount the file system.

▼ How to Add a Client to the Metadata Server

1 Create the mount point for the file system.

For example:

```
# mkdir /sharefs1
```

2 (Optional) Create an `/etc/opt/SUNWsamfs/hosts.file-system-name.local` local hosts configuration file.

You might want to perform this step if your Sun QFS shared host systems have multiple host interfaces. The local hosts configuration file defines the host interfaces that the metadata server

and the client hosts can use when accessing the file system. You use this file to specify how file system traffic should flow over public and private networks in your environment.

The following code example shows a sample local hosts configuration file.

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

For more information on creating the local hosts file, see [“Creating the Local Hosts Configuration File” on page 79](#).

3 If you want to move files from an existing Sun QFS file system into a new Sun QFS shared file system, ensure that each file system to be modified is backed up.

File systems should be backed up regularly according to your site's policies. If you are comfortable with the backup files that already exist for your file systems, you do not need to back them up again now.

4 Unmount the file system.

For instructions, see [“Unmounting File Systems” in *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide*](#).

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

For example:

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

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

The following code example shows a sample file.

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host Host IP Server Not Server
# Name Addresses Priority Used Host
# -----
titan titan-ge0 1 - server
tethys tethys-ge0 2 - server
```

For more information about creating the hosts configuration file, see [“Creating the Shared Hosts File on the Metadata Server” in *Using Sun QFS and Sun Storage Archive Manager with Oracle Solaris Cluster*](#).

Converting a Shared File System to an Unshared File System

Converting a Sun QFS shared file system to an unshared Sun QFS file system requires two tasks:

- Removing the shared clients.
- Converting the metadata server.

This section describes these procedures.

▼ How to Remove a Client From a Shared File System

1 Unmount the file system.

For instructions, see “[Unmounting File Systems](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide*.

2 Delete the file system's entry from the `/etc/opt/SUNWsamfs/mcf` file.

3 Delete the file system's entry from the `/etc/vfstab` file.

4 Notify the `sam-fsd` daemon of the configuration changes by issuing the `samd config` command.

```
# samd config
```

5 Delete the mount point for the file system.

▼ How to Convert a Shared Metadata Server to an Unshared System

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

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

Depending on your software, these files might include the `mcf`, `archiver.cmd`, `defaults.conf`, `samfs.cmd`, and `inquiry.conf` files. Back up these files for all file systems. Also make sure that you have backup copies of files in the `/etc/opt/SUNWsamfs` directory and files in the `/var/opt/SUNWsamfs` directory.

3 If you want to move files from an existing Sun QFS shared file system into a new Sun QFS file system, ensure that each file system to be modified is backed up.

File systems should be backed up regularly according to your site's policies. This is described as the last step in the installation procedure. If you are comfortable with the backup files that already exist for your file systems, you do not need to back them up again now.

4 Unmount the file system.

For instructions, see “[Unmounting File Systems](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide*.

5 Run the `samfsck -F -U file-system-name` command to convert the Sun QFS shared file system to an unshared file system.

For *file-system-name*, specify the name of the Sun QFS shared file system that you are converting to a new unshared file system. For example:

```
# samfsck -F -U samfs1
```

6 In the `/etc/opt/SUNWsamfs/mcf` file, remove the shared keyword from the file system's Additional Parameters field.

For example:

```
# Equipment Eq Eq Family Dev Add
# Identifier Ord Type Set State Params
# -----
samfs1 10 ma samfs1 on
/dev/dsk/c2t50020F23000065EE0s6 11 mm samfs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12 mr samfs1 on
/dev/dsk/c7t50020F2300006099d0s6 13 mr samfs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14 mr samfs1 on
```

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

For example:

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

8 Delete the `/etc/opt/SUNWsamfs/hosts.file-system-name` configuration file.**9 Notify the `sam-fsd` daemon of the configuration changes by issuing the `samd config` command.**

```
# samd config
```

10 Issue the `mount` command to mount the file system.

Client-Server Communications in a Shared File System

The behavior of the shared file system is that of an interruptible hard connection. Each client tries repeatedly to communicate with the metadata server even if the server is unavailable. If the metadata server is not responding, a user can terminate any pending, blocked I/O transmission by pressing Ctrl-C. If the I/O attempt is interrupted, the client persists until the I/O completes.

The system generates the following messages to describe status conditions:

SAM-FS: Shared server is not responding.

This message is also generated if the client `sam-sharefsd` daemon is not active or if the server `sam-sharefsd` daemon is not active. When the server responds, it generates the following message:

SAM-FS: Shared server is responding.

If the file system is not mounted on the metadata server but it is mounted on the client, the system generates the following message:

SAM-FS: Shared server is not mounted.

When the shared file system mounts on the server, it generates the following message:

SAM-FS: Shared server is mounted.

Because the metadata server looks up file names on behalf of all clients, performance can be slow with the default size of the Oracle Solaris directory name lookup cache (DNLC) on the metadata server. To increase performance when clients are frequently opening a large number of files, you might want to double or even triple the size of this cache from its default.

This procedure is documented in the *Oracle Solaris Tunable Parameters Reference Manual*. The parameter that controls the size of the directory name lookup cache is `ncsize`.

Adding Disk Cache to a File System

To increase the disk cache for a file system, you add disk partitions or disk drives, and then update the `mcf` file and use the `samgrowfs` command to expand the file system. You do not need to reinitialize or restore the file system.

When making changes to the `mcf` file, be aware of the following:

- You can configure up to 252 disk partitions in a file system.
- To increase the size of a Sun QFS file system, you must add at least one new metadata partition. Metadata partitions require an Equipment Type value of `mm`.
- If you want to add new partitions for metadata or for data, add them to the `mcf` file after the existing disk partitions.
- Do not change the Equipment Identifier name in the `mcf` file. If the name in the `mcf` file does not match the name in the superblock, the file system can no longer be mounted. Instead, the following message is logged in `/var/adm/messages`:

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

▼ How to Add Disk Cache to a File System

- 1 **Unmount the file system you want to expand. If the file system is shared, unmount the file system on all client hosts and then on the metadata server.**

You can then perform the remaining steps in this procedure on the metadata server. For more information about unmounting a file system, see “[Unmounting File Systems](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide*.

- 2 **If you want to rename the file system during this procedure, use the `samfsck` command with the `-R` and `-F` options.**

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

- 3 **In the `/etc/opt/SUNWsamfs/mcf` file, add the disk cache.**

- 4 **Check for errors in the `mcf` file.**

```
# sam-fsd
```

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

- 5 **Propagate the `mcf` file changes to the system:**

```
# samd config
```

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

- 6 **Issue the `samgrowfs` command on the file system that is being expanded.**

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

- 7 **Mount the file system.**

For information about mounting a Sun QFS file system, see the `mount_samfs(1M)` man page.

- 8 **If the file system is a Sun QFS shared file system, edit the `mcf` file on each participating client host to match the metadata server's `mcf` file.**

Recreating a File System

You must re-create the file system in order to do any of the following tasks:

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

▼ How to Back Up and Re-Create a File System

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

Depending on your software, these files might include the `mcf`, `archiver.cmd`, `defaults.conf`, `samfs.cmd`, or `inquiry.conf` files. Back up these files for all file systems in your Sun QFS environment. Also make sure that you have backup copies of files in the `/etc/opt/SUNWsamfs` directory, files in the `/var/opt/SUNWsamfs` directory, and shared hosts files.

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

File systems should be backed up regularly according to your site's policies. If you are comfortable with the backup files that already exist for your file systems, you do not need to back them up again now. If, however, you need to back up your file systems to preserve information created since the last dump file was created, do so now. For information about how to create a dump file using `qfsdump`, see [Chapter 9, “Backing Up SAM-QFS Data and Files,” in *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide*](#).

3 Unmount the file system.

For instructions, see [“Unmounting File Systems” in *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide*](#)

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

5 In the `/etc/opt/SUNWsamfs/mcf` file, add, change, or remove partitions.

For more information, see [“Adding Disk Cache to a File System” on page 90](#).

6 Check for errors in the `mcf` file.

```
# sam-fsd
```

If the output from this command indicates errors in the `mcf` file, correct them before proceeding to the next step.

7 Propagate the `mcf` file changes to the system.

```
# samd config
```

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

8 Issue the `sammkfs` command to re-create the file system.

For example, the following command creates `samfs10`:

```
# sammkfs samfs10
```

9 Mount the file system.

For information about mounting a Sun QFS file system, see the `mount_samfs(1M)` man page.

10 Change to the mount point of the file system.

11 Use the `qfs restore` command, or use SAM-QFS Manager, to restore each file.

Restore from the existing dump file or from the dump file created in Step 1. For more information, see the `qfsdump(1M)` man page or the SAM-QFS Manager online help.

Administering File System Quotas

This chapter provides information about file system quotas and how to administer the quotas in a Sun QFS file system.

About File System Quotas

File system quotas control the amounts of online and total disk space that can be consumed by a specific user, by a group of users, or by a site-determined group of users called an *admin set*.

Quotas help control the size of a file system by limiting the amount of space and the number of inodes that each user can consume. Quotas can be especially useful on file systems that contain user home directories. After quotas are enabled, you can monitor usage and adjust the quotas as needs change.

A file system provides a user with blocks for data and inodes for files. Each file uses one inode, and file data is stored in a disk allocation unit (DAU). DAU sizes are determined at the time the file system is created. Quotas account for disk usage in multiples of 512 bytes.

Types of Quotas, Quota Files, and Quota Records

You can set quotas according to user ID, group ID, or an administrator's site-specific grouping. This site-specific grouping is called an *admin set ID*. You can use an admin set ID, for example, to identify a collection of users working on a project for which file system quotas are imposed.

Quotas are enabled when the `quota` mount option is in effect and the system detects the presence of one or more quota files in the file system's root directory. The `quota` mount option is enabled by default. If you mount the file system with `noquota` in effect, quotas are disabled. For more information about mount options, see the `mount_samfs(1M)` man page.

Each quota file contains a sequence of records. Record zero is the record for the system administrator's quotas and resource usage. System administrator quotas are never enforced, but

you can use any record, including the system administrator's record, as a template for subsequent records in the quota file. For more information about this practice, see [“How to Enable or Change Limits for Users, Groups, or Admin Sets Using an Existing Quota File”](#) on page 103.

Record one is the record in the quota file for user one, group one, or admin set ID one, depending on the type of quota file. You can edit record one and all subsequent records in order to set different quotas for different users. The following table shows the quota file names and the quotas they enable in /root.

TABLE 6-1 Quota File Names

Quota File Name	Quota Type
.quota_u	UID (system user ID)
.quota_g	GID (system group ID)
.quota_a	AID (system admin set ID)

You can set default quota limits for users by editing record zero in the quota file and having those used as the initial quota settings for all other users. By default, if user quota limits have not been set specifically, the system uses the values in record zero.

Each quota file requires 128 bytes of space. To calculate the necessary size for the initial zero quota file, use the following formula:

$$(*\text{highest-ID} * + 1) \times 128 = \text{xx} / 4096 = \text{zero quota file size}$$

Soft Limits and Hard Limits

You can set both soft and hard limits. A hard limit specifies a fixed amount of system resources available for use, which the system never allows a user to exceed. A soft limit specifies a level of system resource use that can be exceeded temporarily, up to the hard limit. The soft limit is never larger than the hard limit.

If a user attempts to allocate resources beyond the hard limit, the operation fails and generates an EDQUOT error.

After a user exceeds a soft limit, a timer starts, and the user enters a grace period. While the timer is ticking, the user is allowed to operate above the soft limit. After the user goes below the soft limit, the timer is reset. If the grace period ends and the timer stops without the user's having gone below the soft limit, the soft limit is then enforced as a hard limit.

For example, assume that a user has a soft limit of 10,000 blocks and a hard limit of 12,000 blocks. If the user's block usage exceeds 10,000 blocks and the timer exceeds the grace period, this user is no longer able to allocate more disk blocks on that file system until usage drops below the 10,000-block soft limit.

As administrator, you can use the `samquota` command to set the timer value. The `squota` command is a user version of the `samquota` command. The `squota` user command contains options that users can specify to obtain information about quotas that pertain to them.

Disk Blocks and File Limits

A user could exceed an inode quota without using any blocks by creating all empty files. A user could also use only one inode and still exceed the block quota by creating a file that is large enough to consume all data blocks in the user's quota.

File system quotas are expressed in terms of the number of 512-byte blocks that a user can allocate. However, disk space is allocated to user files in terms of DAUs. The DAU setting is specified by the `-a allocation-unit` option to the `sammkfs` command. Setting a block quota to a multiple of the file system DAU is preferable. Otherwise, users can allocate only up to the block count, rounded down to the nearest DAU. See [“Enabling Default Quota Values” on page 102](#) for instructions on setting block quotas.

Enabling Quotas

You can enable quotas through a process that includes editing system files, creating quota files, and entering various quota commands.

Guidelines for Setting Up Quotas

Before you enable quotas, you should determine how much disk space and how many inodes to allocate to each user. If you want to be sure that the total file system space is never exceeded, you can divide the total size of the file system by the number of users. For example, if three users share a 100-megabyte slice and have equal disk space needs, you could allocate 33 megabytes to each. In environments in which not all users are likely to reach their limits, you might want to set individual quotas so that they add up to more than the total size of the file system. For example, if three users shared a 100-megabyte slice, you could allocate 40 megabytes to each.

You can use the following quota commands, in the formats shown, for displaying quota information:

- The `squota` command enables end users to retrieve quota information for themselves on a user, group, or admin set basis.

- The `samquota` command enables system administrators to retrieve quota information or to set quotas.
- Use the `-U` option for a user. For example:


```
# samquota -U janet /mount-point
```
- Use the `-G` option for a group. For example:


```
# samquota -G pubs /mount-point
```
- Use the `-A` option for an admin. For example:


```
# samquota -A 99 /mount-point
```

▼ How to Configure a New File System to Use Quotas

Use this procedure if you are creating a new file system and no files currently reside in the file system. To configure an existing file system to use quotas, see [“How to Configure an Existing File System to Use Quotas” on page 99](#).

Before you start this procedure, make sure that you do not have the `noquota` mount option specified in your `samfs.cmd` or `/etc/vfstab` files.

1 Become superuser.

2 Create the file system.

Either follow the steps outlined in the *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide*, or use the examples in [Chapter 3, “mcf File Examples,”](#) to create the `mcf` file, create the mount point, initialize the file system, and so on.

3 Mount the file system.

For example:

```
# mount /qfs1
```

4 Use the `dd` command to create the quota files.

The arguments to this command depend on the type of quota you are creating, as follows:

- To create admin set quotas:


```
# dd if=/dev/zero of=/qfs1/.quota_a bs=4096 count=1
```
- To create group quotas:


```
# dd if=/dev/zero of=/qfs1/.quota_g bs=4096 count=1
```
- To create user quotas:


```
# dd if=/dev/zero of=/qfs1/.quota_u bs=4096 count=1
```

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

5 Unmount the file system in which the quota files have been created.

For example:

```
# umount /qfs1
```

The file system must be unmounted so it can be remounted and have its quota files read at mount time.

6 Perform a file system check.

In the following example, the `-F` option resets the in-use values in the quota files.

```
# samfsck -F qfs1
```

7 Remount the file system.

The system enables quotas when it detects the presence of one or more quota files in the root directory.

Note – You do not need to include the quota mount option in the `/etc/vfstab` or `samfs.cmd` file. The quota mount option is enabled by default with the `mount` command, and quotas are enabled automatically when the system detects the presence of quota files.

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

8 Use the `samquota` command to set quotas for users, groups, or admin sets.

Subsequent sections in this chapter provide procedures and show examples of this process. For more information about the `samquota` command, see the `samquota(1M)` man page.

▼ How to Configure an Existing File System to Use Quotas

Use this procedure if you are creating quotas for a file system that is already populated with files. If you are configuring a new file system to use quotas, see [“How to Configure a New File System to Use Quotas” on page 98](#).

Before you start this procedure, make sure that you do not have the `noquota` mount option specified in your `samfs.cmd` or `/etc/vfstab` files.

1 Become superuser.**2 Use the `mount` command to examine the `/etc/mnttab` file and ensure that the file system is mounted.**

```
# mount
```

Make sure that the file system is listed in the mount list that is displayed.

3 Change to the root directory of the file system for which quotas are to be enabled.

For example:

```
# cd /oldfs1
```

4 Use the `ls -a` command to retrieve the list of files in this directory and verify that quotas do not already exist on the file system.

If any of the following files are present, quotas have been enabled for this file system: `.quota_u`, `.quota_g`, or `.quota_a`

If any quota type is established for a file system, you can establish any other quota type later. Be careful not to modify existing quota files when adding new ones.

5 If the quota files do not exist for the types of quotas you wish to enforce, use the `dd` command to create the quota files.

Determine the highest existing ID numbers of the types of quotas you want to enforce. Make the initial, zero, quota files large enough to hold the records for those IDs (each quota file record requires 128 bytes).

For example, if you want to enable admin set quotas, and the highest admin set ID in use on the file system is 1024, the calculation is as follows:

$$(1024 + 1) \times 128 = 131200$$

$$131200/4096 = 32.031\dots$$

You would use the following command:

```
# dd if=/dev/zero of=/oldfs1/.quota_a bs=4096 count=33
```

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

6 Unmount the file system in which the quota files have been created.

For example:

```
# umount /oldfs1
```

The file system must be unmounted so it can be remounted and have its quota files read at mount time. For more information about unmounting a file system, see [“Unmounting File Systems” in *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide*](#).

7 Perform a file system check.

This command updates records allocated in the quota files with current usage information, for example:

```
# samfsck -F /oldfs1
```

8 Remount the file system in which the quota files have been created.

The system enables quotas when it detects the presence of one or more quota files in the `/root` directory.

You do not need to include the quota mount option in the `/etc/vfstab` or `samfs.cmd` file. The quota mount option is enabled by default with the `mount` command, and quotas are enabled automatically when the system detects the presence of quota files.

Note – If quota files are present and if the file system is mounted with quotas disabled, the quota records become inconsistent with actual usages when blocks or files are allocated or freed. If a file system with quotas is mounted and run with quotas disabled, run the `samfsck -F` command to update the quota file usage counts before again remounting the file system with quotas enabled.

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

9 Use the `samquota` command to set quotas for users, groups, or admin sets.

Subsequent sections in this chapter provide procedures and show examples of this process. For more information about the `samquota` command, see the `samquota(1M)` man page.

▼ How to Assign Admin Set IDs to Directories and Files

1 Become superuser.

2 Set the admin IDs.

Use the `samchaid` command to change the admin set IDs for the directory or file, as follows:

- To set IDs for a file or directory, specify the directory name or path. For example:

```
# samchaid 100 admin.dir
```

- To set IDs for a directory tree, use the `-R` and (if necessary) the `-h` options. The `-R` option specifies a recursive operation, and the `-h` option changes links, not targets. For example:

```
# samchaid -R -h 22 /qfs1/joe /qfs1/nancee
```

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

Setting Infinite Quotas

Users with infinite quotas are never denied access to any available file system resource. You can set infinite quota values into record zero of the user, group, or admin set ID quota files. You can then use this record as the default value for a new user, group, or admin set ID.

You can use the `samquota` command to set the quota limit to zero.

For example:

```
# samquota -U fred -b 0:h -f 0:h /qfs1
```

You can set infinite quotas for particular users, groups, or admin set IDs by setting zero values for all hard and soft limits. The following example shows how to set infinite quotas.

```
# samquota -G sam -b 0:s,h -f 0:s,h /sam6
# samquota -G sam /sam6
```

	Type	ID	In Use	Online Limits		In Use	Total Limits	
				Soft	Hard		Soft	Hard
/sam6								
Files	group	101	339	0	0	339	0	0
Blocks	group	101	248	0	0	2614	0	0
Grace period				0s			0s	

--> Infinite quotas in effect.

Enabling Default Quota Values

You can use the `samquota` command to enable a default quota for a user, group, or admin set through default limits in user, group, or admin set zero. For example, the following command sets default quotas for all admin set IDs:

```
# samquota -A 0 -b 12000:s -b 15000:h -b 12G:s:t -b 15G:h:t \
-f 1000:s -f 1200:h -t 1w /qfs1
```

This command sets any user's uninitialized admin set quota limits as follows:

- The soft online block limit (`-b limit:s`) is set to 12,000 blocks.
- The hard online block limit (`-b limit:h`) is set to 15,000 blocks.
- The total soft block limit (`-b limit:s:t`) is set to 12 gigablocks.
- The total hard block limit (`-b limit:h:t`) is set to 15 gigablocks.
- The soft file limit (`-f limit:s`) is set to 1000 files.
- The hard file limit (`-f limit:h`) is set to 1200 files.
- The grace period (`-t _limit_`) is set to one week.

Note – If a quota record already exists, the existing values remain in effect. This situation might occur, for example, if the admin group already has blocks assigned to it.

You can set similar default quotas for users or groups by specifying `-U 0` or `-G 0`, respectively, in place of `-A 0`.

Enabling Limits

You can use the `samquota` command to enable a set of limits for a particular user, group, or admin set. For example, the following commands enable limits for users, groups, and admin sets, respectively.

```
# samquota -U joe -b 15000:s -b 20000:h -b 12G:s:t -b 15G:h:t \
-f 500:s -f 750:h -t 3d /qfs1
# samquota -G proj -b 15000:s -b 20000:h -b 12G:s:t -b 15G:h:t \
-f 500:s -f 750:h -t 3d /qfs1
# samquota -A 7 -b 15000:s -b 20000:h -b 12G:s:t -b 15G:h:t \
-f 500:s -f 750:h -t 3d /qfs1
```

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

▼ How to Enable or Change Limits for Users, Groups, or Admin Sets Using an Existing Quota File

After quotas are established, you can use an existing quota file as a template for creating limits for another user, group, or admin set. You can also use this procedure to change any of the quota settings.

1 Retrieve a quota file and direct the output to a temporary file.

Use the `samquota` command with the `-e` option and with one or more of the following options: `-U user-ID`, `-G group-ID`, or `-A adminset-ID`.

The following example shows how to retrieve the `quota.group` file to use as a template.

Note – You can use a group quota entry as a template to create a user quota entry.

```
# samquota -G sam -e /sam6 > /tmp/quota.group
# cat /tmp/quota.group

# Type ID
#
#           Online Limits
#           soft      hard
# Files
# Blocks
# Grace Periods
#
# samquota -G 102 \
-f      200:s:o -f      300:h:o      -f      200:s:t -f      300:h:t \
-b     40000:s:o -b     60000:h:o      -b 40000000:s:t -b 60000000:h:t \
-t      1d:o      -t 1d:t      /sam6
```

2 Save the file and exit the editor.

3 To apply the changes, execute the file using the shell.

For example:

```
# sh -x /tmp/quota.group
```

The `-x` option directs the shell to echo the commands it executes. You can omit the `-x` option if desired.

Checking Quotas

After you have enabled disk and inode quotas, you can check these quotas. The `samquota` command is an administrator command that generates a quota report on an individual user, group, or admin set. The `squota` command is a user command that enables users to check their own individual quotas. You must become superuser to use the `samquota` command.

The `file` argument specifies a file system for the specified user, group, or admin set. The `file` argument can also be the name of any file in the file system. Typically, `file` is the name of the root directory of the file system.

- To display user quotas in effect for mounted file system, using the following command:

```
# samquota -U user-ID [ file ]
```

For `userID`, specify the numeric user ID or user name of the account whose quotas are being examined.

For `file`, specify a file system for the specified user, group, or admin set. The `file` argument can also be the name of any file in the file system. Typically, `file` is the name of the root directory of the file system.

The following example retrieves user `hm1259` quota statistics in the `sam6` file system on the server and displays output indicating that this user is not exceeding the quota.

```
# samquota -U hm1259 /sam6
```

	Type	ID	In Use	Online Limits Soft	Hard	In Use	Total Limits Soft	Hard
/sam6								
Files	user	130959	13	100	200	13	100	200
Blocks	user	130959	152	200	3000	272	1000	3000
Grace period				0s			0s	

The following example retrieves user `memil` quota statistics in all mounted Sun QFS file systems and displays output indicating that this user is exceeding the quota.

The output format for the command is the same for user, groups and admin sets. Note the plus sign (+) in the Blocks row of the output. If the soft quota limit is also being exceeded, a plus sign is also displayed in the Files row.

```
# samquota -U memil
```

	Type	ID	In Use	Online Limits Soft	Hard	In Use	Total Limits Soft	Hard
/sam6								
Files	user	130967	4	500	750	4	500	750
Blocks	user	130967	41016+	40000	50000	41016	50000	50000
Grace period				1w			0s	
--> Warning: online soft limits to be enforced in 6d23h36m45s								
/sam7								
Files	user	130967	4	500	750	4	500	750
Blocks	user	130967	4106	40000	50000	4106	50000	50000
Grace period				1w			0s	

If a hard limit has been exceeded, or if the soft limit has been exceeded and the grace period has expired, the In Use field is marked with an asterisk character (*). If a quota record's limits are inconsistent (for example, if a soft limit is larger than a hard limit), an exclamation point is used to mark the field, and all allocation operations are prevented.

- To display group quotas, use the following command:

```
# samquota -G group-ID [ file ]
```

For *group-ID*, specify the numeric group ID or the group name for the group of users whose quotas are being examined.

For example, the following command retrieves user quota statistics for the group `turtles` in the `qfs3` file system:

```
# samquota -G turtles /qfs3
```

- To display group quotas, use the following command:

```
# samquota -G group-ID [ file ]
```

For *group-ID*, specify the numeric group ID or the group name for the group of users whose quotas are being examined. For *file*, specify a file system for the specified group. The *file* argument can also be the name of any file in the file system. Typically, *file* is the name of the root directory of the file system.

For example, the following command retrieves user quota statistics for the group `turtles` in the `qfs3` file system:

```
# samquota -G turtles /qfs3
```

- Display admin set quotas, using the following command:

```
# samquota -A adminsetID [ file ]
```

For *adminsetID*, specify the numeric admin set ID of the administrator set whose quotas are being examined.

For example, the following command retrieves user quota statistics for the admin set 457 in all mounted file systems:

```
# samquota -A 457
```

Changing and Removing Quotas

You can change quotas to adjust the amount of disk space or number of inodes allocated to users. You can also remove quotas from users or from an entire file system.

▼ How to Change the Grace Period

1 Retrieve quota statistics for a user, group, or admin set.

See “How to Enable or Change Limits for Users, Groups, or Admin Sets Using an Existing Quota File” on page 103 for instructions.

The following example retrieves information about group `sam` and shows that this group is over its soft limit.

```
# samquota -G sam /sam6
```

Type	ID	In Use	Online Limits		In Use	Total Limits		
			Soft	Hard		Soft	Hard	
/sam6								
Files	group	101	32	2000	2000	32	2000	2000
Blocks	group	101	41888*	40000	60000000	43208	60000000	60000000
Grace period				1w			1w	
---> Online soft limits under enforcement (since 30s ago)								

2 Examine the output and determine the new limits.

3 Change the soft time limit grace period. The following example shows the `samquota` command options to use.

```
# samquota -U user-ID -t interval file
# samquota -G group-ID -t interval file
# samquota -A admin-ID -t interval file
```

The arguments for these commands are as follows:

- *user-ID* is the numeric user ID or user name of the user whose quotas are being changed.
- *group-ID* is the numeric group ID or the group name for the group of users whose quotas are being changed.
- *admin-ID* is the numeric admin set ID of the administrator set whose quotas are being changed.
- *interval* is the duration of the grace period. Specify an integer to indicate the quantity and specify a unit of time, if desired. The default unit is `s`, which indicates seconds. You can specify `w` for weeks, `d` for days, `h` for hours, or `m` for minutes.
- *file* is the file system for the specified user, group, or admin set. The *file* argument can also be the name of any file in the file system. Typically, *file* is the name of the root directory of the file system.

Example 6-1 Changing the Grace Period

1. To change the grace period for user `memil`, first verify the quotas.

```
# samquota -U memil /sam6
```

Online Limits

Total Limits

```

Type   ID    In Use   Soft   Hard   In Use   Soft   Hard
/sam6
Files  user  130967    4     500    750     4     500    750
Blocks user  130967  41016+  40000  50000  41016  50000  50000
Grace period                3d                    0s
--> Warning:  online soft limits to be enforced in 2d23h59m7s

```

- Shorten the grace period.

```
# samquota -U memil -t 1d /sam6
```

- Verify the new quotas.

```
# samquota -U memil /sam6
```

Type	ID	In Use	Online Limits		In Use	Total Limits	
			Soft	Hard		Soft	Hard
/sam6							
Files	user	130967	4	500	750	4	500
Blocks	user	130967	41016+	40000	50000	41016	50000
Grace period				1d			0s

```

--> Warning:  online soft limits to be enforced in 23h58m31s

```

Changing the Grace Period Expiration

If a user has exceeded the soft quota limit, changing the grace period itself does not modify the expiration timer of any grace periods that have already started. If the grace period is already in effect, you can use the `samquota` command to modify the grace period in one of the following ways:

- Clear the grace period timer.** The next time the user allocates a file or block while still over a soft limit, the grace period timer is reset and the grace period restarts.

The following example shows the command used to clear the timer so it starts counting the next time a user in group `sam` attempts to allocate a block or file in `/sam6`.

```
# samquota -G sam -x clear /sam6
```

```
Setting Grace Timer:  continue? y
```

```
# samquota -G sam /sam6
```

Type	ID	In Use	Online Limits		In Use	Total Limits	
			Soft	Hard		Soft	Hard
/sam6							
Files	group	101	32	2000	2000	32	2000
Blocks	group	101	41888+	40000	60000000	43208	60000000
Grace period				1w			1w

```

--> Warning:  online soft limits to be enforced in 6d23h59m56s

```

- Reset the grace period timer.** When an expiration period is reset, the timer is reset and the grace period restarts. The following example resets the grace period.

```
# samquota -G sam -x reset /sam6
```

```
Setting Grace Timer:  continue? y
```

```
# samquota -G sam /sam6

      Type   ID   In Use      Online Limits
      /sam6  Soft   Hard      In Use      Total Limits
Files group 101     32      2000     2000     32      2000     2000
Blocks group 101    41888    40000 60000000 43208 60000000 60000000
Grace period                1w                        1w
--> Warning: online soft limits to be enforced in 6d23h59m52s
```

- **Set the grace period to a value.** The timer is set to a value, and it starts counting down immediately from that value. There are no restrictions on this value. The value can be larger than the grace period. The following example sets a very long expiration period.

```
# samquota -G sam -x 52w /sam6

Setting Grace Timer: continue? y

# samquota -G sam /sam6

      Type   ID   In Use      Online Limits
      /sam6  Soft   Hard      In Use      Total Limits
Files group 101     32      2000     2000     32      2000     2000
Blocks group 101    41888+    40000 60000000 43208 60000000 60000000
Grace period                1w                        1w
--> Warning: online soft limits to be enforced in 51w6d23h59m54s
```

- **Expire the grace period timer.** The timer is set to expire immediately. The following example expires the grace period.

```
# samquota -G sam -x expire /sam6

Setting Grace Timer: continue? y

# samquota -G sam /sam6

      Type   ID   In Use      Online Limits
      /sam6  Soft   Hard      In Use      Total Limits
Files group 101     32      2000     2000     32      2000     2000
Blocks group 101    41888    40000 60000000 43208 60000000 60000000
Grace period                1w                        1w
--> Online soft limits under enforcement (since 6s ago)
```

Inhibiting Additional File System Resource Allocations

When the file system detects that quota values are not consistent for a user, group, or admin set, it prevents that user, group, or admin set from using any more system resources. You can inhibit file system resource allocations by creating inconsistent quota values. For example, you can inhibit further allocation if the hard block or file limits are lower than the soft block or file limits, or if a user's soft limit is larger than the user's hard limit.

The file system treats an inconsistent quota setting as a special quota. You can set inconsistent quota values into record zero of the user, group, or admin set ID quota files. They can then become the default values for new users, groups, or admin set IDs.

▼ How to Inhibit Additional File System Resource Allocations

1 Become superuser.

2 Obtain, save, and examine current quota information.

The following example shows how to retrieve current group quota information for group `sam` and write it to a backup file.

```
# samquota -G sam -e /sam6 | & tee restore.quota.sam

# Type ID
#
# Online Limits
# soft hard Total Limits
# soft hard
# Files
# Blocks
# Grace Periods
# samquota -G 101 \
-f 2000:s:o -f 2000:h:o -f 2000:s:t -f 2000:h:t \
-b 40000:s:o -b 60000000:h:o -b 60000000:s:t -b 60000000:h:t \
-t 1w:o -t 1w:t \
-x 51w6d23h59m:o -x clear /sam6
```

To obtain quota information about a user quota, specify the `-U user-ID` option in place of the `-G` option. To obtain quota information about an admin set quota, specify the `-A admin-ID` option in place of the `-G` option.

3 Use the `samquota` command to set soft quotas to nonzero quotas and hard quotas to zero quotas.

The following command sets the quotas for group `sam` to be inconsistent.

```
# samquota -G sam -f 1:s -f 0:h -b 1:s -b 0:h /sam6
```

To make the quotas for users or admin sets inconsistent, specify the `-U userID` or `-A adminID` option in place of the `-G` option.

4 Verify your changes, as in the following example:

```
# samquota -G sam /sam6

Online Limits
Type ID In Use Soft Hard In Use Soft Hard
/sam6
Files group 101 32! 1 0 32! 1 0
Blocks group 101 41888! 1 0 43208! 1 0
Grace period 1w 1w

--> Quota values inconsistent; zero quotas in effect.
```

In this output, a zero quota is in effect. The exclamation point characters (!) indicate the over-quota condition in the output.

5 Restore the group's quota to what it was before the file/block allocation was inhibited and then to verify the changed quotas.

The following example shows these commands.

```
# sh restore_quota.sam
Setting Grace Timer: continue? y
Setting Grace Timer: continue? y
# samquota -G sam /sam6
Online Limits
Type ID In Use Soft Hard In Use Soft Hard
/sam6
Files group 101 32 2000 2000 2000 32 2000 2000
Blocks group 101 41888+ 40000 60000000 43208 60000000 60000000
Grace period 1w 1w
---> Warning: online soft limits to be enforced in 6d23h59m54s
```

To perform this operation on a user quota, specify the `-U user-ID` option in place of the `-G` option. To perform this operation on an admin set quota, specify the `-A admin-ID` option in place of the `-G` option.

▼ How to Remove the Quotas for a File System

To remove or disable quotas for a file system, disable quotas in the mount process.

1 Become superuser.

2 (Optional) Add the `noquota` mount option to the `/etc/vfstab` or `samfs.cmd` file.

As an alternative, you can specify `noquota` as an option when you issue the mount command. See Step 4.

3 If the file system is mounted, use the `umount` command to unmount the file system. For example:

```
# umount /myfs
```

For more information, see [“Unmounting File Systems” in Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide](#).

4 Remount the file system.

If you did not perform Step 2, include the `noquota` option with the mount command. For example:

```
# mount -o noquota /myfs
```

5 Manage the quota files by doing one of the following:

- If you expect to reinstate the quota feature later and therefore do not want to destroy the quota files, unmount the file system, run the `samfsck` command with its `-F` option on the file system, and remount the file system with the `noquota` option removed.
- If you do not expect to reinstate the quota feature or if you want to reclaim the space consumed by the quota files, use the `rm` command to remove the `.quota_u`, `.quota_g`, and `.quota_a` files. For example:

```
# rm /myfs/.quota_[agu]
```

▼ How to Correct Quotas

1 Become superuser.

2 If the file system is mounted, unmount the file system.

For example:

```
# umount /myfs
```

If you have difficulty unmounting the file system, see “Unmounting File Systems” in *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide*.

3 Perform a file system check.

The `samfsck` command updates records allocated in the quota files with correct, current usage information. For example:

```
# samfsck -F myfs
```

4 Remount the file system.

For example:

```
# mount /myfs
```


Advanced File System Topics

This chapter discusses advanced topics that are beyond the scope of basic system administration and usage.

Using Daemons, Processes, and Tracing

Understanding system daemons and processes is useful when you are debugging. This section describes the Sun SAM and Sun QFS daemons and processes. It also provides information about daemon tracing.

Daemons and Processes

All daemons are named in the form `sam-daemon_named`. Processes are named in a similar manner, except that processes do not end in the lowercase letter `d`.

The following table shows some of the daemons and processes that can run on your system. Others, such as `sam-genericd` and `sam-catserverd`, might also be running, depending on system activities.

TABLE 7-1 Daemons and Processes

Process	Description
<code>sam-amld</code>	Initializes the Sun SAM automated library daemons: <code>sam-catserverd</code> , <code>sam-scannerd</code> , and <code>sam-robotsd</code> .
<code>sam-archiverd</code>	Automatically archives Sun SAM files. This process runs as long as the Sun SAM file system is mounted.
<code>sam-catserverd</code>	Keeps track of media in Sun SAM and SAM-QFS library catalogs.
<code>sam-fsd</code>	Master daemon.

TABLE 7-1 Daemons and Processes (Continued)

Process	Description
sam-rftd	Transfers data between multiple Sun SAM host systems.
sam-robotd	Starts and monitors automated library media changer control daemons.
sam-scannerd	Monitors all manually mounted removable media devices. The scanner periodically checks each device for inserted archive media cartridges.
sam-sharefsd	Invokes the Sun QFS shared file system daemon.
sam-releaser	Attempts to release disk space occupied by previously archived files on Sun SAM file systems until a low-water mark is reached. The releaser is started automatically when a high-water mark is reached on disk cache and stops when it has finished releasing files. This is a process, not a daemon.
sam-stagealld	Controls the associative staging of Sun SAM files.
sam-stagerd	Controls the staging of Sun SAM files.
sam-rpcd	Controls the remote procedure call (RPC) application programming interface (API) server process.

When you run the software, SMF starts the `sam-fsd` daemon. It should restart automatically in case of failure.

In a shared file system, a `sam-fsd` daemon is always active. In addition, one `sam-sharefsd` daemon is active for each mounted shared file system.

When a shared file system is mounted, the software starts a shared file system daemon (`sam-sharefsd`). TCP sockets are used to communicate between the server and client hosts. All clients that connect to the metadata server are validated against the hosts file. See the `hosts.fs(4)` man page for more information about the hosts file.

The `sam-sharedfsd` daemon on the metadata server opens a listener socket on the port named `sam-qfs`. During the Sun QFS installation process, the `sam-qfs` entry is automatically added to `/etc/services` file. Do not remove this entry. In addition, the shared file system port is defined in the `/etc/inet/services` file as port number 7105. Verify that this port does not conflict with another service.

All metadata operations, block allocation and deallocation, and record locking are performed on the metadata server. The `sam-sharefsd` daemon does not keep any information. Hence, it can be stopped and restarted without causing any consistency problems for the file system.

Trace Files

Several processes can write messages to trace files. These messages contain information about the state and progress of the work performed by the daemons. The messages are primarily used by Oracle support staff to improve performance and diagnose problems. The message content and format are subject to change from release to release.

Trace files can be used in debugging. By default, trace files are not enabled. You can enable trace files by editing the `defaults.conf` file. You can enable tracing for all processes, or you can enable tracing for individual processes. For information about the processes that you can trace, see “`defaults.conf(4)`” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

By default, trace files are written to the `/var/opt/SUNWsamfs/trace` directory. In that directory, the trace files are named for the processes (`archiver`, `catserver`, `fsd`, `ftpd`, `recycler`, `sharefsd`, and `stager`). You can change the names of the trace files by specifying directives in the `defaults.conf` configuration file. You can also set a limit on the size of a trace file and rotate your tracing logs. For information about controlling tracing, see “`defaults.conf(4)`” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

Trace File Content

Trace file messages contain the time and source of the message. The messages are produced by events in the processes. You can select the events by using directives in the `defaults.conf` file.

The default events are as follows:

- Customer notification `syslog` or `notify` file messages
- Nonfatal program errors
- Fatal `syslog` messages
- Process initiation and completion
- Other miscellaneous events

You can also trace the following events:

- Memory allocations
- Interprocess communication
- File actions
- Operator messages
- Queue contents when changed
- Other miscellaneous events

The default message elements (program name, process ID (PID), and time) are always included and cannot be excluded. Optionally, the messages can also contain the following elements:

- The date. (The time is always included.)
- The source file name and line number.
- The event type.

Trace File Rotation

To prevent trace files from growing indefinitely, the `sam-fsd` daemon monitors the size of the trace files and periodically executes the following command:

```
/opt/SUNWsamfs/sbin/trace_rotate
```

This script moves the trace files to sequentially numbered copies. You can modify this script to suit your operation. Alternatively, you can provide this function using `cron` or some other facility.

Determining Which Processes Are Being Traced

To determine which processes are being traced currently, issue the `sam-fsd` command at the command line. The following example shows the output from this command.

EXAMPLE 7-1 `sam-fsd` Command Output

```
# sam-fsd
Trace file controls:
sam-amld /var/opt/SUNWsamfs/trace/sam-amld
      cust err fatal misc proc date
      size 10M age 0
sam-archiverd /var/opt/SUNWsamfs/trace/sam-archiverd
      cust err fatal ipc misc proc queue date module
      size 10M age 0
sam-catsserverd /var/opt/SUNWsamfs/trace/sam-catsserverd
      cust err fatal misc proc date
      size 10M age 0
sam-dbupd /var/opt/SUNWsamfs/trace/sam-dbupd
      cust err fatal misc proc date
      size 10M age 0
sam-fsalogd /var/opt/SUNWsamfs/trace/sam-fsalogd
      cust err fatal misc proc date
      size 10M age 0
sam-fsd /var/opt/SUNWsamfs/trace/sam-fsd
      cust err fatal misc proc date
      size 10M age 0
sam-rftd /var/opt/SUNWsamfs/trace/sam-rftd
      cust err fatal misc proc date
      size 10M age 0
sam-recycler /var/opt/SUNWsamfs/trace/sam-recycler
      cust err fatal ipc misc proc date module type
      size 10M age 0
sam-nrecycler /var/opt/SUNWsamfs/trace/sam-nrecycler
      cust err fatal misc proc date
      size 10M age 0
sam-sharefsd /var/opt/SUNWsamfs/trace/sam-sharefsd
      cust err fatal misc proc date
      size 10M age 0
sam-stagerd /var/opt/SUNWsamfs/trace/sam-stagerd
      cust err fatal ipc misc proc date module
      size 10M age 0
sam-serverd /var/opt/SUNWsamfs/trace/sam-serverd
      cust err fatal misc proc date
```

EXAMPLE 7-1 sam-fsd Command Output (Continued)

```

size      10M  age 0
sam-clientd /var/opt/SUNWsamfs/trace/sam-clientd
cust err fatal misc proc date
size      10M  age 0
fsmgmt    /var/opt/SUNWsamfs/trace/fsmgmt
cust err fatal misc proc date
size      10M  age 0
sam-shrink /var/opt/SUNWsamfs/trace/sam-shrink
cust err fatal misc proc date
size      10M  age 0
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
#

```

For more information about enabling trace files, see “defaults.conf(4)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual* and “sam-fsd(1M)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

Using the set fa Command to Set File Attributes

Sun QFS file systems enable end users to set performance attributes for files and directories. Applications can enable these performance features on a per-file or per-directory basis. You can use these features as follows:

- To select file attributes for files and directories
- To specify the allocation method for the file
- To preallocate file space
- To specify the disk stripe width

For more information about implementing these features, see “setfa(1)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

Selecting File Attributes for Files and Directories

The set fa command sets attributes on a new or existing file. The file is created if it does not already exist.

You can set attributes on a directory as well as a file. When using the set fa command with a directory, files and directories created within that directory inherit the attributes set in the original directory. To reset attributes on a file or directory to the default, use the -d (default) option. When the -d option is used, attributes are first reset to the default and then other attributes are processed.

Preallocating File Space

An end user can preallocate space for a file. This space is associated with a file so that no other files in the file system can use the disk addresses allocated to this file. Preallocation ensures that space is available for a given file, which avoids a file-system-full condition. Preallocation is assigned at the time of the request rather than when the data is actually written to disk.

Note that space can be wasted by preallocation of files. If the file size is less than the allocation amount, the kernel allocates space to the file from the current file size up to the allocation amount. When the file is closed, space below the allocation amount is not freed.

You can preallocate space for a file by using the `setfa` command with either the `-L` or the `-l` (lowercase letter L) option. Both options accept a file length as their argument. Use the `-L` option for an existing file, which can be empty or contain data. Use the `-l` option for a file that has no data yet. If you use the `-l` option, the file cannot grow beyond its preallocated limit.

For example, to preallocate a 1-gigabyte file named `/qfs/file_alloc`, type the following:

```
# setfa -l 1g /qfs/file_alloc
```

After space for a file has been preallocated, truncating a file to 0 length or removing the file returns all space allocated for a file. There is no way to return only part of a file's preallocated space to the file system. In addition, if a file is preallocated with the `-l` option, there is no way to extend the file beyond its preallocated size in future operations.

Selecting a File Allocation Method and Stripe Width

By default, a file uses the allocation method and stripe width specified at mount time. See [“mount_samfs\(1M\)” in Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual](#). However, an end user might want to use a different allocation scheme for a file or directory. The user could do this by using the `setfa` command with the `-s` (stripe) option.

The allocation method can be either round-robin or striped. The `-s` option specifies the allocation method and the stripe width, as shown in the following table.

TABLE 7-2 File Allocations and Stripe Widths

-s Option	Allocation Method	Stripe Width	Explanation
0	Round-robin	Not applicable	The file is allocated on one device until that device has no space.
1-255	Striped	1-255 DAUs	The file is striped across all disk devices with this number of DAUs per disk.

The following example shows how to create a file explicitly by specifying a round-robin allocation method:

```
# setfa -s 0 /qfs/100MB.rrobin
```

The following example shows how to create a file explicitly by specifying a striped allocation method with a stripe width of 64 DAUs (preallocation is not used):

```
# setfa -s 64 /qfs/file.stripe
```

Selecting a Striped Group Device

A user can specify that a file begin allocation on a particular striped group. If the file allocation method is round-robin, the file is allocated on the designated stripe group.

The following example shows `setfa` commands specifying that `file1` and `file2` be independently spread across two different striped groups.

EXAMPLE 7-2 `setfa` Commands to Spread Files Across Striped Groups

```
# setfa -g0 -s0 file1
# setfa -g1 -s0 file2
```

This capability is particularly important for applications that must achieve levels of performance that approach raw device speeds. For more information, see “[setfa\(1\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

Accommodating Large Files

When manipulating very large files, pay careful attention to the size of disk cache that is available on the system. If you try to write a file that is larger than your disk cache, behavior differs depending on the type of file system that you are using:

- If you are using a non-archiving file system, the system returns an ENOSPC error.
- If you are using an archiving file system, the program blocks, waiting for space that might never exist because the available disk space is insufficient to handle the request.

If you are operating within an archiving environment and your application must write a file that is larger than the disk cache, you can segment the file with the `segment` command. For more information about the `segment` command, see “[segment\(1\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual* or see “[Using Segmented Files](#)” in *Sun Storage Archive Manager 5.3 Configuration and Administration Guide*.

Configuring a Multireader File System

A multireader file system consists of a single writer host and multiple reader hosts. The `writer` and `reader` mount options that enable the multireader file system are compatible with Sun QFS file systems only. The mount options are described in this section and in the `mount_samfs(1M)` man page.

You can mount the multireader file system on the single writer host by specifying the `-o writer` option with the `mount` command. The host system with the `writer` mount option is the only host system that is allowed to write to the file system. The `writer` host system updates the file system. You must ensure that only one host in a multireader file system has the file system mounted with the `writer` mount option enabled. If `-o writer` is specified, directories are written through to disk at each change and files are written through to disk at close.



Caution – The multireader file system can become corrupted if more than one writer host has the file system mounted at one time. The site administrator's responsibility should ensure that this situation does not occur.

You can mount a multireader file system on one or more reader hosts by specifying the `-o reader` option with the `mount` command. There is no limit to the number of host systems that can have the multireader file system mounted as a reader.

A major difference between the multireader file system and a Sun QFS shared file system is that the multireader host reads metadata from the disk, and the client hosts of a Sun QFS shared file system read metadata over the network. The Sun QFS shared file system supports multireader hosts. In this configuration, multiple shared hosts can be adding content while multiple reader hosts are distributing content.

Note – You cannot specify the `writer` option on any host if you are mounting the file system as a Sun QFS shared file system. You can, however, specify the `reader` option. If you want a Sun QFS shared file system client host to be a read-only host, mount the Sun QFS shared file system on that host with the `reader` mount option. In addition, set the `sync_meta` mount option to 1 if you use the `reader` option in a Sun QFS shared file system. For more information about the Sun QFS shared file system, see [Chapter 5, “Configuring a Shared File System.”](#) For more information about mount options, see “`mount_samfs(1M)`” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

You must ensure that all readers in a multireader file system have access to the device definitions that describe the `ma` device. Copy the lines from the `mcf` file that resides on the primary metadata server host to the `mcf` files on the alternate metadata servers. After copying the lines, you might need to update the information about the disk controllers because, depending on your configuration, disk partitions might not show up the same way across all hosts.

In a multireader file system environment, the Sun QFS software ensures that all servers accessing the same file system can always access the current environment. When the writer closes a file, the Sun QFS file system immediately writes all information for that file to disk. A reader host can access a file after the file is closed by the writer. You can specify the `refresh_at_eof` mount option to help ensure that no host system in a multireader file system gets out of sync with the file system.

By default, the metadata information for a file on a reader host is invalidated and refreshed every time a file is accessed. If the data changes, it is invalidated. This policy includes any type of access, whether through `cat`, `ls`, `touch`, `open`, or other methods. This immediate refresh rate ensures that the data is correct at the time the refresh is done, but it can affect performance. Depending on your site preferences, you can use the `mount` command's `-o invalid=n` option to specify a refresh rate between 0 seconds and 60 seconds. If the refresh rate is set to a small value, the Sun QFS file system reads the directory and other metadata information *n* seconds after the last refresh. More frequent refreshes result in more overhead for the system, but stale information can exist if *n* is nonzero.



Caution – If a file is open for a read on a reader host, that file could be removed or truncated by the writer. Use another mechanism, such as application locking, to protect the reader from inadvertent writer actions.

About I/O Types

This section provide information about paged I/O, direct I/O, and switching between the I/Os.

Paged I/O

When paged I/O is used, user data is cached in virtual memory pages, and the kernel writes the data to disk. The standard Oracle Solaris OS interfaces manage paged I/O. Paged I/O (also called buffered or cached I/O) is selected by default.

Direct I/O

Direct I/O is a process by which data is transferred directly between the user's buffer and the disk. This means that much less time is spent in the system. For performance purposes, specify direct I/O only for large, block-aligned, sequential I/O.

The `setfa` command and the `sam_setfa` library routine both have a `-D` option that sets the direct I/O attribute for a file or directory. If applied to a directory, files and directories created in that directory inherit the direct I/O attribute. After the `-D` option is set, the file uses direct I/O.

You can also select direct I/O for a file by using the Oracle Solaris OS `directio` function call. If you use the function call to enable direct I/O, the setting lasts only while the file is active.

To enable direct I/O on a file system basis, do one of the following:

- Specify the `-o forcedirectio` option with the `mount` command.
- Put the `forcedirectio` keyword in the mount option column of the `/etc/vfstab` file, or use it as a directive in the `samfs.cmd` file.

For more information, see the `setfa(1)`, `sam_setfa(3)`, `samfs.cmd(4)`, `directio(3C)`, and `mount_samfs(1M)` man pages in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

I/O Switching

By default, paged I/O is performed and I/O switching is disabled. However, the Sun QFS file systems support automatic I/O switching, a process by which a site-defined amount of paged I/O occurs before the system switches automatically to direct I/O.

I/O switching should reduce page cache usage on large I/O operations. To enable I/O switching, use `samu(1M)`, or use the `dio_wr_consec` and `dio_rd_consec` parameters as directives in the `samfs.cmd` file or as options with the `mount` command.

For more information about these options, see “`mount_samfs(1M)`” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual* or “`samfs.cmd(4)`” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

SMB Service in SAM-QFS

This chapter provides information about the properties that support the Oracle Solaris Server Message Block (SMB) service in the Sun QFS and SAM-QFS software. This feature provides the ability to share files between Oracle Solaris and Windows systems.

The following new SAM-QFS properties facilitate sharing files between these systems by using the Oracle Solaris SMB service.

- Case-insensitivity support
- DOS attributes support
- Sun QFS NFSv4 ACL support

Note the following basic requirements for supporting the SMB service in SAM-QFS:

- Oracle Solaris 11 OS
- The file system has to support NFSv4 ACLs. For information about creating a file system with NFSv4 ACLs or converting the existing POSIX ACLs to NFSv4, see [“Using ACLs to Protect Sun QFS and SAM-QFS Files” on page 126](#).

Note – The Oracle Solaris OS provides a Server Message Block (SMB) protocol server and client implementation that includes support for numerous SMB dialects including NT LM 0.12 and Common Internet File System (CIFS). The terms CIFS and SMB can be considered interchangeable.

For information about integrating an Oracle Solaris SMB server into an existing Windows environment and mounting SMB shares on Oracle Solaris systems, see [Oracle Solaris Administration: SMB and Windows Interoperability](#).

Getting Started With the Oracle Solaris SMB Service

Oracle Solaris SMB services can operate in two modes: domain and workgroup. These modes are mutually exclusive. Choose one or the other based on your environment and authentication needs.

- If you have an Active Directory (AD) domain and want to give domain users access to the Oracle Solaris SMB service, choose domain mode by joining that domain.
- If you have no AD domains or have no need to support domain users and you want to use local Oracle Solaris users to access the SMB service, choose workgroup mode by joining the workgroup.

After you have successfully joined a workgroup or an AD domain, you can create and access SMB shares.

For information about configuring the Oracle Solaris SMB server, see “[Configuring the SMB Server Operation Mode \(Task Map\)](#)” in *Oracle Solaris Administration: SMB and Windows Interoperability*.

share Command

The share command enables you to manage SMB shares on various file system types. For more information, see the share(1M) man page.

Use the following syntax to enable a SMB share on a SAM-QFS file system:

```
# share -F smb /sqfs1
```

Case-Insensitivity Support

Traditionally, UNIX and POSIX file systems have case-sensitive file names. SAM-QFS file systems already support the case-sensitive behavior. The SMB service requires case-insensitive behavior. In order to support the SMB service in SAM-QFS, a mixed-mode behavior has been implemented. The mixed-mode behavior indicates that the file system can support requests for both case-sensitive and case-insensitive matching behavior.

When a case-insensitive matching request is made of a mixed-sensitivity file system, the behavior is generally the same as would be expected of a purely case-insensitive file system. The difference is that a mixed-sensitivity file system might contain files or directories with multiple names that are unique from a case-sensitive perspective but not unique from the case-insensitive perspective.

For example, a directory might contain files foo, Foo, and F00. If a request is made to case-insensitively match any of the possible forms of foo (for example foo, FOO, FoO, fOo, and so on), one of the three existing files is chosen as the match by the matching algorithm. The file

that the algorithm chooses as a match is not guaranteed, but the same file will be chosen as a match for any of the forms of foo. The file chosen as a case-insensitive match for foo, FOO, foO, Foo, and so on, is always the same as long as the directory remains unchanged.

DOS Attributes Support

In order to completely support SMB service, Sun QFS has to support certain file attributes. These attributes can be set and cleared by the owner of a file or a user or a group that has been granted the permission using the `write_attributes` ACE permission.

The following table lists the file attributes that are supported in the Sun OFS file system.

TABLE 8-1 File Attributes Supported in the Sun QFS File System

Attribute	Description
<code>appendonly</code>	Marks a file so that it can only be appended to.
<code>archive</code>	Marks files for archiving. It is set when a file is created or modified. This attribute is currently not used by SAM-QFS.
<code>createtime</code>	Timestamp when a file is created. The owner of the file or any user with <code>write_attributes</code> permission can modify this value to any time.
<code>hidden</code>	Marks a file as hidden. The file appears transparent in a Windows system.
<code>immutable</code>	Prevents the content of a file from being modified or deleted. Also, prevents all metadata changes except for access time updates. When this attribute is placed on a directory, it will prevent the deletion and creation of files and directories. All attempts to modify the content of a file or directory marked as immutable will fail with an error. All attempts to modify any attributes (with the exception of access time) of a file marked as immutable will fail with an error.
<code>nodump</code>	Oracle Solaris systems have no special semantics for this attribute.
<code>nounlink</code>	Prevents a file from being deleted. On a directory, the attribute will also prevent any changes to the contents of the directory. That is, no files within the directory can be removed or renamed. All attempts to unlink or rename files and directories that are marked as <code>nounlink</code> will return an error.
<code>offline</code>	Indicates whether a file is offline. If a file is released in the SAM-QFS file system, then the offline attribute is set. Windows systems will not attempt to preview files that have this attribute set.
<code>readonly</code>	Marks a file as readonly. Once a file is marked as readonly, the content data of the file cannot be modified. Other metadata for the file can still be modified. This attribute can be set on directories but it has no semantic meaning. All attempts to modify the content of the file will return an error.

TABLE 8-1 File Attributes Supported in the Sun QFS File System (Continued)

Attribute	Description
sparse	Available to users and applications to indicate that a file can be interpreted as sparse. It does not indicate whether the file is actually sparse and it has no special semantics on the Oracle Solaris OS. The sparse attribute will be cleared if the file is truncated to zero length.
system	Marker for special files on Windows systems.

EXAMPLE 8-1 Viewing System Attributes Using the `ls` Command

You can use the `ls` command to view the attributes of a SAM-QFS file. For example:

```
# ls -/v file1
-rw-r--r-- 1 root root 40560 Jun 28 11:54 file1
{archive,nohidden,noreadonly,nosystem,noappendonly,
nonodump,noimmutable,nonounlink,nooffline,sparse}
```

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

EXAMPLE 8-2 Modifying System Attributes Using the `chmod` Command

You can use the `chmod` command to modify the system attributes of a SAM-QFS file. For example, to modify a file to be read-only:

```
# chmod S+vreadonly file2
# ls -/v file2
-rw-r--r-- 1 root root 12 Jun 28 11:56 file2
{noarchive,nohidden,readonly,nosystem,noappendonly,
nonodump,noimmutable,nonounlink,offline,nosparse}
```

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

Using ACLs to Protect Sun QFS and SAM-QFS Files

Previous versions of Oracle Solaris supported an ACL implementation that was primarily based on the POSIX-draft ACL specification. The POSIX-draft based ACLs are used to protect UFS files and are translated by versions of NFS prior to NFSv4.

With the introduction of NFSv4, a new Oracle Solaris ACL model fully supports the interoperability that NFSv4 offers between UNIX and non-UNIX clients. The new ACL implementation, as defined in the NFSv4 specification, provides much richer semantics that are based on NT-style ACLs.

The main differences of the new ACL model are as follows:

- Based on the NFSv4 specification and similar to NT-style ACLs.
- Provide much more granular set of access privileges. For more information, see [Table 8-3](#).

- Set and displayed with the `chmod` and `ls` commands rather than the `setfacl` and `getfacl` commands.
- Provide richer inheritance semantics for designating how access privileges are applied. For more information, see [“ACL Inheritance” on page 131](#).

Both ACL models provide more fine-grained access control than is available with the standard file permissions. Much like POSIX-draft ACLs, the new ACLs are composed of multiple Access Control Entries (ACEs).

POSIX-draft style ACLs use a single entry to define what permissions are allowed and what permissions are denied. The new ACL model has two types of ACEs that affect access checking: ALLOW and DENY. As such, you cannot infer from any single ACE that defines a set of permissions whether the permissions not defined in that ACE are allowed or denied.

For more information about the new Oracle Solaris ACL model, see [“New Solaris ACL Model” in *Oracle Solaris Administration: ZFS File Systems*](#).

Mapping User and Group Identities in SAM-QFS

Currently, SAM-QFS does not support ephemeral IDs or SIDs. Therefore, all Windows identities must be explicitly defined using the `idmap` service or must be provided by the active directory service. Identities encountered by the SMB server that do not have an explicit mapping defined will be defaulted automatically to the nobody identity.

For more information about identity mapping administration, see [Chapter 2, “Identity Mapping Administration \(Tasks\),” in *Oracle Solaris Administration: SMB and Windows Interoperability*](#).

Creating and Converting File Systems to Support NFSv4 ACLs

POSIX ACLs are the default ACLs present in the SAM-QFS file systems. In order to support the SMB service, SAM-QFS file systems also have to support NFSv4 ACLs.

- Use the `sammkfs -A` command to create a file system with NFSv4 ACLs. For example, to create a file system, `sqfs1`, with NFSv4 ACLs:

```
# sammkfs -A -S sqfs1
```

For more information about the `-A` option in the `sammkfs` command, see [“sammkfs\(1M\)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*](#).

- Use the `samfsck -A` command to convert existing POSIX ACLs to NFSv4 ACLs. For example, to convert from POSIX ACLS to NFSv4 ACLs on an existing file system, `sqfs2`:

```
# samfsck -F -A sqfs2
```

Note – The conversion from POSIX ACLs to NFSv4 cannot be reversed. The conversion applies only to file systems versions V2 or V2A.

For more information about the -A option in the `samfsck` command, see “[samfsck\(1M\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

Syntax Descriptions for Setting ACLs

Two basic ACL formats are provided as follows:

Syntax for Setting Trivial ACLs

```
chmod [options] A[index]{+|=}owner@ |group@
|everyone@:access-permissions/...[:inheritance-flags]:deny | allow file
```

```
chmod [options] A-owner@, group@,
everyone@:access-permissions/...[:inheritance-flags]:deny | allow file ...
```

```
chmod [options] A[index]- file
```

Syntax for Setting Non-Trivial ACLs

```
chmod [options]
A[index]{+|=}user|group:name:access-permissions/...[:inheritance-flags]:deny | allow file
```

```
chmod [options] A-user|group:name:access-permissions/...[:inheritance-flags]:deny |
allow file ...
```

```
chmod [options] A[index]- file
```

owner@, group@, everyone@

Identifies the *ACL-entry-type* for trivial ACL syntax. For a description of *ACL-entry-types*, see [Table 8-2](#).

user or group:*ACL-entry-ID=username or groupname*

Identifies the *ACL-entry-type* for explicit ACL syntax. The user and group *ACL-entry-type* must also contain the *ACL-entry-ID*, *username* or *groupname*. For a description of *ACL-entry-types*, see [Table 8-2](#).

access-permissions/.../

Identifies the access permissions that are granted or denied. For a description of ACL access privileges, see [Table 8-3](#).

inheritance-flags

Identifies an optional list of ACL inheritance flags. For a description of the ACL inheritance flags, see [Table 8-4](#).

deny | allow

Identifies whether the access permissions are granted or denied.

In the following example, the *ACL-entry-ID* value is not relevant.

```
group@:write_data/append_data/execute:deny
```

The following example includes an *ACL-entry-ID* because a specific user (*ACL-entry-type*) is included in the ACL.

```
0:user:gozer:list_directory/read_data/execute:allow
```

When an ACL entry is displayed, it looks similar to the following:

```
2:group@:write_data/append_data/execute:deny
```

The **2** or the *index-ID* designation in this example identifies the ACL entry in the larger ACL, which might have multiple entries for owner, specific UIDs, group, and everyone. You can specify the *index-ID* with the `chmod` command to identify which part of the ACL you want to modify. For example, you can identify index ID 3 as **A3** to the `chmod` command, similar to the following:

```
chmod A3=user:venkman:read_acl:allow filename
```

ACL entry types, which are the ACL representations of owner, group, and other, are described in the following table.

TABLE 8-2 ACL Entry Type

ACL Entry Type	Description
owner@	Specifies the access granted to the owner of the object.
group@	Specifies the access granted to the owning group of the object.
everyone@	Specifies the access granted to any user or group that does not match any other ACL entry.
user	With a user name, specifies the access granted to an additional user of the object. Must include the <i>ACL-entry-ID</i> , which contains a <i>username</i> or <i>user-ID</i> . If the value is not a valid numeric UID or <i>username</i> , the ACL entry type is invalid.
group	With a group name, specifies the access granted to an additional group of the object. Must include the <i>ACL-entry-ID</i> , which contains a <i>groupname</i> or <i>group-ID</i> . If the value is not a valid numeric GID or <i>groupname</i> , the ACL entry type is invalid.

ACL access privileges are described in the following table.

TABLE 8-3 ACL Access Privileges

Access Privilege	Description
add_file	Permission to add a new file to a directory.
add_subdirectory	On a directory, permission to create a subdirectory.
delete	Permission to delete a file.
delete_child	Permission to delete a file or directory within a directory.
execute	Permission to execute a file or search the contents of a directory.
list_directory	Permission to list the contents of a directory.
read_acl	Permission to read the ACL (ls).
read_attributes	Permission to read basic attributes (non-ACLs) of a file. (Think of basic attributes as the stat level attributes). Allowing this access mask bit means the entity can execute <code>ls(1)</code> and <code>stat(2)</code> .
read_data	Permission to read the contents of the file.
read_xattr	Permission to read the extended attributes of a file or perform a lookup in the file's extended attributes directory.
write_xattr	Permission to create extended attributes or write to the extended attributes directory. Granting this permission to a user means that the user can create an extended attribute directory for a file. The attribute file's permissions control the user's access to the attribute.
write_data	Permission to modify or replace the contents of a file.
write_attributes	Permission to change the times associated with a file or directory to an arbitrary value.
write_acl	Permission to write the ACL or the ability to modify the ACL by using the <code>chmod</code> command.
write_owner	Permission to change the file's owner or group, or, the ability to execute the <code>chown</code> or <code>chgrp</code> commands on the file. Permission to take ownership of a file or permission to change the group ownership of the file to a group of which the user is a member. If you want to change the file or group ownership to an arbitrary user or group, then the <code>PRIV_FILE_CHOWN</code> privilege is required.

EXAMPLE 8-3 Modifying Trivial ACLs on SAM-QFS Files

In the following example, a trivial ACL exists on `file.1`:

EXAMPLE 8-3 Modifying Trivial ACLs on SAM-QFS Files (Continued)

```
# ls -v file.1
-rw-r--r-- 1 root root 206674 Jun 14 10:54 file.1
0:owner@:read_data/write_data/append_data/read_xattr/write_xattr
/read_attributes/write_attributes/read_acl/write_acl/write_owner
/synchronize:allow
1:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
2:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
:allow
```

ACL Inheritance

The purpose of using ACL inheritance is so that a newly created file or directory can inherit the appropriate ACLs without disregarding the existing permission bits on the parent directory.

By default, ACLs are not propagated. If you set a non-trivial ACL on a directory, it is not inherited to any subsequent directory. You must specify the inheritance of an ACL on a file or directory.

The optional inheritance flags are described in the following table.

TABLE 8-4 ACL Inheritance Flags

Inheritance Flag	Description
<code>file_inherit</code>	Only inherit the ACL from the parent directory to the directory's files.
<code>dir_inherit</code>	Only inherit the ACL from the parent directory to the directory's subdirectories.
<code>inherit_only</code>	Inherit the ACL from the parent director. This option applies only to newly created files or subdirectories and not to the directory itself. This flag requires the <code>file_inherit</code> flag, the <code>dir_inherit</code> flag, or both, to indicate what to inherit.
<code>no_propagate</code>	Only inherit the ACL from the parent directory to the first-level contents of the directory, not the second-level or subsequent contents. This flag requires the <code>file_inherit</code> flag, the <code>dir_inherit</code> flag, or both, to indicate what to inherit.

By default, ACLs are not propagated through a directory structure.

EXAMPLE 8-4 Granting Default ACL Inheritance

In the following example, a non-trivial ACE of `read_data/write_data/execute` is applied for user `gozer` on `test.dir`.

```
# chmod A+user:gozer:read_data/write_data/execute:allow test.dir
# ls -dv test.dir
drwxr-xr-x+ 2 root root 2 Jun 15 10:40 test.dir
```

EXAMPLE 8-4 Granting Default ACL Inheritance (Continued)

```

0:user:gozer:list_directory/read_data/add_file/write_data/execute:allow
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/read_xattr/write_xattr/execute/read_attributes
  /write_attributes/read_acl/write_acl/write_owner/synchronize:allow
2:group@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow

```

If a `test.dir` subdirectory is created, the ACE for user `gozer` is not propagated. User `gozer` would only have access to `sub.dir` if the permissions on `sub.dir` granted him access as the file owner, group member, or `everyone@`.

```

# mkdir test.dir/sub.dir
# ls -dv test.dir/sub.dir
drwxr-xr-x  2 root  root          2 Jun 15 10:41 test.dir/sub.dir
0:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/read_xattr/write_xattr/execute/read_attributes
  /write_attributes/read_acl/write_acl/write_owner/synchronize:allow
1:group@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
2:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow

```

EXAMPLE 8-5 Granting ACL Inheritance on Files Within Directories

In the following example, certain permissions are applied to all the newly created files in a directory but the permissions do not apply to the directory itself. The `file_inherit` flag indicates that the permissions are for the files, and the `inherit_only` flag indicates that the permissions are for inheritance and will not apply to the directory itself.

```

# chmod A+user:bob:read_data/execute:file_inherit/inherit_only:deny mydir
# ls -vd mydir
dr-xr-xr-x+  2  root  root          4096 Jul  5 19:10 mydir
0:user:bob:list_directory/read_data/execute:file_inherit/inherit_only:deny
1:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/read_xattr/write_xattr/execute/delete_child
  /read_attributes/write_attributes/read_acl/write_acl/write_owner
  /synchronize:allow
2:group@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow
3:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
  /read_acl/synchronize:allow

```

If a `myfile` file is created in the `mydir` directory, the `myfile` file will inherit all the access permissions automatically.

```

# cd mydir
# touch myfile
# ls -v myfile
-r--r--r--+  1  root  root          0 Jul  5 19:11 myfile

```

EXAMPLE 8-5 Granting ACL Inheritance on Files Within Directories *(Continued)*

```
0:user:bob:read_data/execute:file_inherit/inherit_only:deny
1:owner@:read_data/write_data/append_data/read_xattr/write_xattr
  /read_attributes/write_attributes/read_acl/write_acl/write_owner
  /synchronize:allow
2:group@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
3:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize:allow
```


Configuring WORM-FS File Systems

This chapter provides information about the WORM technology and how to configure tunable WORM-FS file systems.

About WORM-FS File Systems

Write-once read-many (WORM) technology is used in many applications for data integrity reasons and because of the accepted legal admissibility of stored files that use the technology.

Note – The `SUNWsamfswm` package has been merged with the `SUNWqfs` and `SUNWsamfs` packages. You do not have to install the `SUNWsamfswm` package separately to enable the WORM-FS functionality.

To enable the WORM-FS functionality, specify the mount options from the following list:

- `worm_capable`
- `worm_lite`
- `worm_emul`
- `emul_lite`

For more information about the mount options, see [“Enabling the WORM-FS Feature” on page 136](#).

The WORM-FS feature offers default and customizable file-retention periods, data and path immutability, and subdirectory inheritance of the WORM setting.

WORM-FS can operate in one of two modes:

- Sun standard compliance mode (referred to as standard mode), which is the default

- Sun emulation compliance mode (referred to as emulation mode), which is designed to provide compatibility with the emulation mode of the StorageTek 5320 network attached storage (NAS) appliance and is similar to an interface defined by Network Appliance

One difference between standard and emulation mode is a restriction on the nature of files that can be retained. Specifically, in standard mode, files with any UNIX executable permissions cannot be retained. Emulation mode has no such restriction. The restriction in standard mode exists because the retention trigger defined for NFS and FTP specifying that the `setuid` mode be set on the file. Once a file is retained, a client will see the `setuid` mode bit set. However, the restriction on executable files will prevent the possible security hole of allowing an executable file owned by the root user to be made WORM and therefore impossible to remove. A benefit of this approach is that the user or application can more easily determine which files on the system are WORM-protected files.

Using WORM-FS With NFS Clients

If you are using WORM-FS on Oracle Solaris 10 or later with NFS clients connected to it, ensure that NFS version 4 is enabled on the NFS clients and sever.

If you are running an older version of the Oracle Solaris OS (before Oracle Solaris 10) and NFS version 3, the NFS client might not show the WORM-FS files. Add the following line to the `/etc/system` file and then reboot the system:

```
set nfs:nfs_allow_preepoch_time = 1
```

Enabling the WORM-FS Feature

The following table shows the four mount options that you can use to enable the WORM-FS feature.

Option	Brief Description	Notes
<code>worm_capable</code>	Standard WORM mode	The WORM trigger command, <code>chmod 4000 filename/directory-name</code> , is used to set the WORM bit on a file or directory.
<code>worm_lite</code>	Relaxes some of the standard WORM mode restrictions	The system administrator is allowed to delete files before retention expiration and reduce the file retention period. File data and path integrity remain immutable. See “WORM Lite Options” on page 138 for more information.

Option	Brief Description	Notes
worm_emul	WORM emulation mode, which is designed to provide compatibility with the emulation mode of the StorageTek 5320 network attached storage (NAS) appliance	Provides standard WORM functionality with a different WORM trigger. The WORM bit is set by changing a directory or file from writeable to read-only.
emul_lite	WORM emulation mode, which is designed to provide compatibility with the "lite" version of the StorageTek 5320 network attached storage (NAS) appliance	Provides standard WORM lite functionality with a different WORM trigger. The WORM bit is set by changing a directory or file from writeable to read-only. As with the worm_lite option, the administrator can carry out special operations on files. See "WORM Lite Options" on page 138 for more information.

These four mount options are somewhat exclusive. You can upgrade from "lite" to standard WORM mode, but you cannot change from standard WORM mode to emulation mode, or from emulation to standard mode. These options can be provided on the command line when the file system is mounted, listed in `/etc/vfstab`, or provided in `/opt/SUNWsamfs/samfs.cmd`. The normal rules of precedence for mount options apply.

The WORM attribute is stored in the mount table and enables WORM files to be created in directories anywhere in the file system.

Note – You must have system administration privileges to set a WORM mount option in `/etc/vfstab`.

The following example shows a WORM-FS mount option. The file system `samfs1` mounted at `/samfs1` is WORM-capable and has the default retention period for files set to 60 minutes.

EXAMPLE 9-1 Using WORM-FS Mount Options

```
# cat /etc/vfstab
#device device mount FS fsck mount mount
#to mount to fsck point type pass at boot options
#
fd - /dev/fd fd - no -
/proc - /proc proc - no -
/dev/dsk/c0t0d0s1 - - swap - no -
samfs1 - /samfs1 samfs - yes worm_capable,def_retention=60
swap - /tmp tmpfs - yes -
```

After the WORM-FS feature has been enabled and at least one WORM file is resident in the file system, the file system's superblock is updated to reflect the WORM capability. Any subsequent attempt to rebuild the file system through `samkfs` will fail unless you are using the `worm_lite`

EXAMPLE 9-1 Using WORM-FS Mount Options (Continued)

or `emul_lite` mount option.

WORM Lite Options

The `worm_lite` and `emul_lite` mount options create a modified WORM environment that eases the restrictions on actions that can be taken on WORM-enabled volumes and retained files. The WORM lite options can be a solution for companies with document management and retention policies requiring data retention guarantees but not the strict constraints that WORM places on systems. Mechanisms exist to alter and even reverse some data retention decisions.

The WORM lite options can also be used for testing and configuring WORM systems and applications before upgrading to the more strict standard WORM policies.

The WORM lite environment behaves similarly to the standard WORM mode. File data and path remain immutable, but the system administrator is allowed to carry out the following special actions:

- Remove WORM files before the retention time has expired
- Shorten the retention time on WORM files
- Delete WORM Lite-enabled volumes or rebuild them using the `sammkfs` command

Creating WORM Files

A WORM mount option enables a file system to contain WORM files, but it does not automatically create WORM files. To create a WORM file, you must first make the directory WORM-capable. Create directory and then use a WORM trigger command to set the WORM bit on the directory. Depending on the mount option being used, the following WORM trigger commands are available:

- Use `chmod 4000 directory-name` to set the WORM bit if you are using the `worm_capable` or `worm_lite` mount option.
- Remove the write permissions on the directory to set the WORM bit if you are using the `worm_emul` or `emul_lite` mount option.

After setting the WORM bit on a directory, you can create files in that directory and then use the appropriate WORM trigger to set the WORM bit on files that you want retained. The WORM trigger is the same for both files and directories.

Use care when applying the WORM trigger. The file data and path cannot be changed after the file has the WORM feature applied. Once this feature is applied to a file, it is irrevocable. Further, once the WORM trigger is applied to a file, its volume also become a WORM volume

and remains that way. The volume can only be destroyed using a volume management or RAID interface. If one of the WORM lite options was used to create it, the volume can also be rebuilt by using `sammkfs`.

The following examples illustrate using the WORM trigger for each of the four mount options using the system-wide default retention value.

EXAMPLE 9-2 `chmod 4000` WORM Trigger Using Standard WORM Functionality

This example shows a simple application of the WORM trigger `chmod 4000` using standard WORM functionality.

```
[host1]# grep -i worm /etc/vfstab
samfs1 -      /samfs1 samfs -      no      bg,worm_capable

[host1]# cd /samfs1
[host1]# mkdir WORM
[host1]# chmod 4000 WORM
[host1]# sfs -D

WORM:
mode: drwxr-xr-x  links:  2  owner: root      group: root
length: 4096  admin id:  0  inode:  1025.1
access:  Jan 30 15:50  modification: Jan 30 15:50
changed:  Jan 30 15:50  attributes:  Jan  1 1970
creation: Jan 30 15:50  residence:   Jan 30 15:50
worm-capable      retention-period: 0y, 30d, 0h, 0m

[host1]# cd WORM
[host1]# touch test
[host1]# chmod 4000 test
[host1]# sfs -D

test:
mode: -r-Sr--r--  links:  1  owner: root      group: root
length: 0  admin id:  0  inode:  1026.3
access:  Jan 30 15:51  modification: Jan 30 15:51
changed:  Jan 30 15:51  retention-end: Mar  1 15:51 2007
creation: Jan 30 15:51  residence:   Jan 30 15:51
retention: active      retention-period: 0y, 30d, 0h, 0m

[host1]# rm test
rm: test: override protection 444 (yes/no)? yes
rm: test not removed: Read-only file system
[host1]# ls
test
```

EXAMPLE 9-3 `chmod 4000` WORM Trigger Using WORM Lite Functionality

This example shows a simple application of the WORM trigger `chmod 4000` using standard WORM lite functionality.

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 -      /samfs1 samfs -      no      bg,worm_lite
```

EXAMPLE 9-3 `chmod 4000 WORM Trigger Using WORM Lite Functionality` (Continued)

```
[host1]# mount samfs1
[host1]# cd /samfs1
[host1]# mkdir WORM
[host1]# chmod 4000 WORM
[host1]# sfs -D

WORM:
mode: drwxr-xr-x links: 2 owner: root group: root
length: 4096 admin id: 0 inode: 1025.1
access: Jan 30 16:12 modification: Jan 30 16:12
changed: Jan 30 16:12 attributes: Jan 1 1970
creation: Jan 30 16:12 residence: Jan 30 16:12
worm-capable retention-period: 0y, 30d, 0h, 0m

[host1]# cd WORM
[host1]# touch test
[host1]# chmod 4000 test
[host1]# sfs -D
test:
mode: -r-Sr--r-- links: 1 owner: root group: root
length: 0 admin id: 0 inode: 1026.1
access: Jan 30 16:13 modification: Jan 30 16:13
changed: Jan 30 16:13 retention-end: Mar 1 16:13 2007
creation: Jan 30 16:13 residence: Jan 30 16:13
retention: active retention-period: 0y, 30d, 0h, 0m

[host1]# rm test
rm: test: override protection 444 (yes/no)? yes
[host1]# ls
[host1]#
```

EXAMPLE 9-4 `chmod -w WORM Trigger Using WORM Emulation Mode`

This example show a simple application of the WORM trigger using WORM emulation mode.

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 - /samfs1 samfs - no bg,worm_emul

[host1]# mount samfs1
[host1]# cd /samfs1
[host1]# mkdir WORM
[host1]# chmod -w WORM
[host1]# sfs -D

WORM:
mode: drwxr-xr-x links: 2 owner: root group: root
length: 4096 admin id: 0 inode: 1025.1
access: Jan 30 16:26 modification: Jan 30 16:26
changed: Jan 30 16:26 attributes: Jan 1 1970
creation: Jan 30 16:26 residence: Jan 30 16:26
worm-capable retention-period: 0y, 30d, 0h, 0m

[host1]# cd WORM
[host1]# touch test
[host1]# chmod -w test
```

EXAMPLE 9-4 `chmod -w` WORM Trigger Using WORM Emulation Mode (Continued)

```
[host1]# sls -D

test:
  mode: -r--r--r--  links: 1  owner: root    group: root
  length:           0  admin id: 0  inode: 1026.1
  access:           Jan 30 16:26  modification: Jan 30 16:26
  changed:          Jan 30 16:26  retention-end: Mar 1 16:26 2007
  creation:         Jan 30 16:26  residence:     Jan 30 16:26
  retention:        active         retention-period: 0y, 30d, 0h, 0m

[host1]# rm test
rm: test: override protection 444 (yes/no)? yes
rm: test not removed: Read-only file system
[host1]# ls
test
```

EXAMPLE 9-5 `chmod -w` WORM Trigger Using WORM Emulation Lite Mode

This example shows a simple application of the WORM trigger using WORM emulation lite mode.

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 - /samfs1 samfs - no bg,emul_lite

[host1]# mount samfs1
[host1]# cd /samfs1
[host1]# mkdir WORM
[host1]# chmod -w WORM
[host1]# sls -D

WORM:
  mode: drwxr-xr-x  links: 2  owner: root    group: root
  length:          4096  admin id: 0  inode: 1025.1
  access:          Jan 30 16:36  modification: Jan 30 16:36
  changed:         Jan 30 16:36  attributes:   Jan 1 1970
  creation:        Jan 30 16:36  residence:    Jan 30 16:36
  worm-capable     retention-period: 0y, 30d, 0h, 0m

[host1]# cd WORM
[host1]# touch test
[host1]# chmod -w test
[host1]# sls -D

test:
  mode: -r--r--r--  links: 1  owner: root    group: root
  length:           0  admin id: 0  inode: 1026.1
  access:           Jan 30 16:36  modification: Jan 30 16:36
  changed:          Jan 30 16:36  retention-end: Mar 1 16:36 2007
  creation:         Jan 30 16:36  residence:     Jan 30 16:36
  retention:        active         retention-period: 0y, 30d, 0h, 0m

[host1]# rm test
rm: test: override protection 444 (yes/no)? yes
[host1]# ls
[host1]#
```

Retention Periods

The WORM-FS feature also includes file-retention periods that can be customized. Assigning a retention period to a file maintains the WORM features in that file for the specified period of time.

Note – Retention periods cannot extend beyond 01/18/2038 when initially assigning or extending the period using Oracle Solaris/UNIX utilities. These utilities use signed 32-bit numbers to represent time in seconds. Time is measured from the epoch which is January 1, 1970. 2^{31} seconds from the epoch extends to 01/18/2038 around 10:14 PM. You can, however, exceed this date using a default retention period. See [“Setting the Default Retention Period” on page 144](#).

Do one of the following to set a retention period for a file:

- Advance the file's access time using the touch utility, or with a program using the libc subroutine utimes(). The file's retention period is stored in minutes. After the access time is advanced, use the appropriate WORM trigger to set the WORM bit.
- Use the default retention period for a file by applying the appropriate WORM trigger. The file system will apply the default retention period. For more information, see [“Setting the Default Retention Period” on page 144](#).

The following example shows the creation of a file in a WORM-capable directory, using the WORM trigger on the file (with the chmod 4000 command), and using the sfs command to display the file's WORM features. This example uses the default retention period of the file system (60 minutes, as set in [Example 9-1](#)).

EXAMPLE 9-6 Creation of a WORM-Capable Directory and WORM File

```
# cd WORM
# echo "This is a test file" >> test
# sfs -D
test:
    mode: -rw-r--r-- links: 1 owner: root group: other
    length: 20 admin id: 0 inode: 1027.1
    access: Oct 30 02:50 modification: Oct 30 02:50
    changed: Oct 30 02:50 attributes: Oct 30 02:50
    creation: Oct 30 02:50 residence: Oct 30 02:50

    checksum: gen no_use not_val algo: 0

# chmod 4000 test
# sfs -D
test:
    mode: -r--r--r-- links: 1 owner: root group: other
    length: 20 admin id: 0 inode: 1027.1
    access: Oct 30 02:50 modification: Oct 30 02:50
    changed: Oct 30 02:50 retention-end: Oct 30 2005 03:50
    creation: Oct 30 02:50 residence: Oct 30 02:50
```

EXAMPLE 9-6 Creation of a WORM-Capable Directory and WORM File (Continued)

```
retention: active retention-period: 0y, 0d, 1h, 0m
checksum: gen no_use not_val algo: 0
```

With the addition of the WORM-FS feature, three states are possible for a file in a Sun QFS file system:

- Normal
- Retained
- Expired

The normal state represents the state of an ordinary file in a Sun QFS file system. A transition to the retained, or *active*, state occurs when the WORM bit is set on a file. The expired, or *over*, state occurs when the file's retention period is exceeded.

When a retention period is assigned to a file and the WORM trigger is applied to it, the file's path and data are immutable. When the retention period expires, the state is changed to expired but the path and data remain immutable.

When a file is in an expired state, only two operations are available:

- Extension of the retention period. (The retention period cannot be shortened unless you are using a `WORM lite` option.)
- Deletion of the file.

If the retention period is extended, the file's state returns to active and the new end date and duration are set accordingly.

Both hard and soft links to files can be used with the WORM-FS feature. Hard links can be established only with files that reside in a WORM-capable directory. After a hard link is created, it has the same WORM characteristics as the original file. Soft links can also be established, but a soft link cannot use the WORM features. Soft links to WORM files can be created in any directory in a Sun QFS file system.

Another attribute of the WORM-FS feature is directory inheritance. New directories created under a directory that includes a WORM attribute inherit this attribute from their parent. If a directory has a default retention period set, this retention period is also inherited by any new subdirectories. The WORM bit can be set on any file whose parent directory is WORM-capable. Users can set the WORM feature on directories and files that they own or have access to by using normal UNIX permissions.

Note – A WORM-capable directory can only be deleted if it contains no WORM files.

Setting the Default Retention Period

The default retention period for a file system can be set as a mount option in the `/etc/vfstab` file. For example:

```
samfs1 - /samfs1 samfs - nobg,worm_emul,def_retention=1y60d
```

The format for setting the default retention period is `MyNdOhPm`, in which *M*, *N*, *O* and *P* are non-negative integers and *y*, *d*, *h* and *m* stand for years, days, hours, and minutes respectively. Any combination of these units can be used. For example, `1y5d4h3m` indicates 1 year, 5 days, 4 hours, and 3 minutes; `30d8h` indicates 30 days and 8 hours and `300m` indicates 300 minutes. This format is backward compatible with software versions prior to 4U5, in which the retention period was specified in minutes. Although the granularity of the period is in minutes, the accuracy of the period is based on one day. Also, the function that handles days, hours, and minutes does not account for leap years when determining retention periods.

You can also use the default retention period to set a file or directory's retention period beyond the year 2038. Set the default retention period to a value which exceeds 2038 and mount the file system. Then use the appropriate WORM trigger to apply the default retention period. The following example shows how to use the default retention period to set a retention period on a directory and file which exceeds the year 2038.

EXAMPLE 9-7 Extending the Retention Period Beyond 2038

```
[root@ns-east-44]# grep samfs1 /etc/vfstab
samfs1 - /samfs1 samfs - no
bg,worm_capable,def_retention=34y
[root@ns-east-44]# mount samfs1
[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod 4000 WORM
[root@ns-east-44]# sfs -D
WORM:
mode: drwxr-xr-x links: 2 owner: root group: root
length: 4096 admin id: 0 inode: 1026.1
access: Feb 20 14:24 modification: Feb 20 14:24
changed: Feb 20 14:24 attributes: Jul 26 1970
creation: Feb 20 14:24 residence: Feb 20 14:24
worm-capable retention-period: 34y, 0d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod 4000 test
[root@ns-east-44]# sfs -D
test:
mode: -r-Sr--r-- links: 1 owner: root group: root
length: 0 admin id: 0 inode: 1027.1
access: Feb 20 14:24 modification: Feb 20 14:25
changed: Feb 20 14:25 retention-end: Feb 20 14:25 2041
creation: Feb 20 14:24 residence: Feb 20 14:24
retention: active retention-period: 34y, 0d, 0h, 0m
```

You can also set a default retention period for a directory using the touch utility, as described in the next section. This retention period overrides the default retention period for the file system and is inherited by any subdirectories.

Setting the Retention Period Using touch

You can use the touch utility to set or extend a file's or directory's retention period. You can also use touch to shorten the default retention period for a directory (but not for a file).

To set the retention period, you must first advance the file's or directory's access time using touch, and then apply the WORM trigger by using the chmod command or removing write permissions, depending on the WORM mode in place at the time.

The following example shows the use of the touch utility to set a file's retention period, followed by the application of the WORM trigger.

EXAMPLE 9-8 Using touch and chmod to Set the Retention Period

```
# touch -a -t200508181125 test
# sfs -D
test:
    mode: -rw-r--r-- links: 1 owner: root group: root
    length: 0 admin id: 0 inode: 1027.1
    access: Aug 18 2005 modification: Aug 18 11:19
    changed: Aug 18 11:19 attributes: Aug 18 11:19
    creation: Aug 18 11:19 residence: Aug 18 11:19

# chmod 4000 test
# sfs -D
test:
    mode: -r-Sr--r-- links: 1 owner: root group: root
    length: 0 admin id: 0 inode: 1027.1
    access: Aug 18 2005 modification: Aug 18 11:19
    changed: Aug 18 11:19 retention-end: Aug 18 2005 11:25
    creation: Aug 18 11:19 residence: Aug 18 11:19
    retention: active retention-period: 0y, 0d, 0h, 6m
```

The -a option for touch changes the access time of the file or directory. The -t option specifies the time to be used for the access time field. The format for the time argument is [[CC]YY]MMDDhhmm[.SS], as follows:

- [CC] – The first two digits of the year.
- [YY] – The second two digits of the year.
- MM – The month of the year (01–12).
- DD – The day of the month (01–31).
- hh – The hour of the day (00–23).
- mm – The minute of the hour (00–59).
- [SS] – The second of the minute (00–61).

The CC, YY, and SS fields are optional. If CC and YY are not given, the default is the current year. See the touch(1) man page for more information about these options.

To set the retention period to permanent retention, set the access time to its largest possible value: 203801182214.07.

The following example shows how to use touch to extend a file's retention period.

EXAMPLE 9-9 Using touch to Extend a File's Retention Period

```
# sls -D test
test:
    mode: -r-Sr--r-- links: 1 owner: root group: root
    length: 0 admin id: 0 inode: 1029.1
    access: Aug 18 11:35 modification: Aug 18 11:33
    changed: Aug 18 11:33 retention-end: Aug 18 2005 11:35
    creation: Aug 18 11:33 residence: Aug 18 11:33
    retention: over retention-period: 0y, 0d, 0h, 2m
# touch -a -t200508181159 test
# sls -D
test:
    mode: -r-Sr--r-- links: 1 owner: root group: root
    length: 0 admin id: 0 inode: 1029.1
    access: Aug 18 11:35 modification: Aug 18 11:33
    changed: Aug 18 11:33 retention-end: Aug 18 2005 11:59
    creation: Aug 18 11:33 residence: Aug 18 11:33
    retention: active retention-period: 0y, 0d, 0h, 26m
```

In this example, the retention period is extended to Aug 18, 2005 at 11:59AM, which is 26 minutes from the time the WORM trigger was initially applied.

Note – Using touch to extend the retention period is independent of the active WORM mode.

Using sls to View WORM-FS Files

Use the sls command to view WORM file attributes. The -D option shows whether a directory is WORM-capable. Use this option on a file to display when the retention period began, when it will end, the current retention state, and the duration as specified on the command line.

The retention period start time and duration (in minutes) are stored in the file's inode.

To access this information directly you must use a program similar to the following example.

EXAMPLE 9-10 Program for Direct Access to Retention Period Start Time and Duration

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
```

EXAMPLE 9-10 Program for Direct Access to Retention Period Start Time and Duration *(Continued)*

```

/*
 * SAMFS includes in /opt/SUNWsamfs/include
 */
#include "stat.h"
#include "lib.h"

/*
 * SAMFS libraries in /opt/SUNWsamfs/lib
 */
int
main(int argc, char **argv)
{
    char ibuf[1000];
    struct sam_stat buf;

    if (argc != 2) {
        printf("usage: sam_worm_stat filename\n");
        exit(-1);
    }

    if (sam_stat(argv[1], &buf, sizeof(buf)) == 0) {
        (void)time_string(buf.rperiod_start_time,
buf.rperiod_start_time, ibuf);
        printf("retention period start is %s\n", ibuf);
        printf("retent period duration is %ld minutes\n",
buf.rperiod_duration);
    } else {
        printf("can not sam_worm_stat %s\n", argv[1]);
    }
}

```

The following example shows how `sls -D` displays a file's retention status.

EXAMPLE 9-11 Using `sls` to Find a File's Retention Status

```

# sls -D test
test:
    mode: -r-Sr--r-- links: 1 owner: root group: root
    length: 5 admin id: 0 inode: 1027.1
    access: Aug 18 2008 modification: Aug 18 11:19
    changed: Aug 18 11:19 retention-end: Aug 18 2008 11:25
    creation: Aug 18 11:19 residence: Aug 18 11:19
    retention: active retention-period: 0y, 0d, 0h, 6m

```

In this example, the retention state is active, as shown by the `retention: active` designation, meaning that the file has the WORM bit set. The retention period started on August 18, 2008, at 11:19 and will end on August 18, 2008, at 11:25. The retention period was specified to be 0 years, 0 days, 0 hours, and 6 minutes.

Using `sfind` to Find WORM-FS Files

Use the `sfind` utility to search for files that have certain retention periods. See “`sfind(1)`” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual* for more information on the options. The following options are available:

- `-ractive` — Finds files whose retention period is active.
- `-rover` — Finds files whose retention periods have expired.
- `-rafter date` — Finds files whose retention period will end after the specified date. The date is specified as `YYYYMMDDHHmm`, where `YYYY` is the year, `MM` is the month, `DD` is the day, `HH` is the hour, and `mm` is minutes. The following example shows the command to find files whose retention period expires after 12/24/2004 at 15:00.


```
# sfind -rafter 200412241500
```
- `-rremain time` — Finds files that have retention periods with at least the specified amount of time left. The time is specified as `MyNdOhPm`, where `M`, `N`, `O`, and `P` are arbitrary non-negative integers and `y`, `d`, `h`, and `m` represent the number of years, days, hours, and minutes, respectively. For example, the following command finds files for which more than 1 year, 10 days, 5 hours, and 10 minutes remain before expiration.


```
# sfind -rremain 1y10d5h10m
```
- `-rlonger time` — Finds files that have retention periods longer than the specified amount of time. The time is specified as `MyNdOhPm`, where `M`, `N`, `O`, and `P` are arbitrary non-negative integers and `y`, `d`, `h`, and `m` represent the number of years, days, hours, and minutes, respectively. For example, the following command finds files that have retention periods longer than 10 days.


```
# sfind -rlonger 10d
```
- `-rpermanent` — Finds files whose retention period is permanent.

Tunable Parameters

This chapter describes various ways that you can improve file system and archiving performance.

Note – Oracle recommends that you experiment with performance tuning outside of a production environment. Tuning variables incorrectly can have unexpected effects on the overall system.

If your site has a support contract with Oracle, please inform the Oracle support personnel if you change performance tuning parameters.

Increasing File Transfer Performance for Large Files

Sun QFS file systems are tuned to work with a mix of file sizes. You can increase the performance of disk file transfers for large files by enabling file system settings.

▼ How to Increase File Transfer Performance

1 Set the maximum device read/write directive.

The `maxphys` parameter in the Oracle Solaris `/etc/system` file controls the maximum number of bytes that a device driver reads or writes at any one time. The default value for the `maxphys` parameter can differ depending on the level of the Oracle Solaris OS, but it is typically around 128 kilobytes.

Add the following line to `/etc/system` to set `maxphys` to 1 megabytes:

```
set maxphys = 0x100000
```

Note – The `maxphys` value *must* be set to a power of two.

2 Set the SCSI disk maximum transfer parameter.

The `sd` driver enables large transfers for a specific file by looking for the `sd_max_xfer_size` definition in the `/kernel/drv/sd.conf` file. If this definition does not exist, the driver uses the value defined in the `sd` device driver definition, `sd_max_xfer_size`, which is 1024 x 1024 bytes.

To enable and encourage large transfers, add the following line at the end of the `/kernel/drv/sd.conf` file:

```
sd_max_xfer_size=0x800000;
```

3 Set the fibre disk maximum transfer parameter.

The `ssd` driver enables large transfers for a specific file by looking for the `ssd_max_xfer_size` definition in the `/kernel/drv/ssd.conf` file. If this definition does not exist, the driver uses the value defined in the `ssd` device driver definition, `ssd_max_xfer_size`, which is 1024 x 1024 bytes.

Add the following line at the end of the `/kernel/drv/ssd.conf` file:

```
ssd_max_xfer_size=0x800000;
```

Note – On Oracle Solaris 10 x86 platforms, this change is made in the `/kernel/drv/sd.conf` file. For a maximum transfer size of 8 Mbytes, the following line is added:

```
sd_max_xfer_size=0x800000
```

4 Reboot the system.

5 Set the writebehind parameter.

This step affects paged I/O only.

The `writebehind` parameter specifies the number of bytes that are written behind by the file system when paged I/O is being performed on a Sun QFS file system. Matching the `writebehind` value to a multiple of the RAID's read-modify-write value can increase performance.

This parameter is specified in units of kilobytes and is truncated to an 8-kilobyte multiple. If set, this parameter is ignored when direct I/O is performed. The default `writebehind` value is 512 kilobytes. This value favors large-block, sequential I/O.

Set the `writebehind` size to a multiple of the RAID 5 stripe size for both hardware and software RAID-5. The RAID-5 stripe size is the number of data disks multiplied by the configured stripe width.

For example, assume that you configure a RAID-5 device with three data disks plus one parity disk (3+1) with a stripe width of 16 kilobytes. The `wri tebehind` value should be 48 kilobytes, 96 kilobytes, or some other multiple, to avoid the overhead of the read-modify-write RAID-5 parity generation. For Sun QFS file systems, the DAU (`sammkfs -a` command) should also be a multiple of the RAID-5 stripe size. This allocation ensures that the blocks are contiguous.

You should test the system performance after resetting the `wri tebehind` size. The following example shows testing timings of disk writes:

```
# timex dd if=/dev/zero of=/sam/myfile bs=256k count=2048
```

You can set the `wri tebehind` parameter from a mount option, from within the `samfs . cmd` file, from within the `/etc/vfstab` file, or from a command within the `samu` utility. For information about enabling this from a mount option, see the `-o wri tebehind=n` option on the `mount_samfs(1M)` man page. For information about enabling this from the `samfs . cmd` file, see “`samfs.cmd(4)`” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*. For information about enabling this from within `samu(1M)`, see “`samu(1M)`” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

6 Set the `readahead` parameter. This step affects paged I/O only.

The `readahead` parameter specifies the number of bytes that are read ahead by the file system when paged I/O is being performed on a Sun QFS file system. This parameter is specified in units of kilobytes and is truncated to an 8-kilobyte multiple. If set, this parameter is ignored when direct I/O is performed.

Increasing the size of the `readahead` parameter increases the performance of large file transfers, but only to a point. You should test the performance of the system after resetting the `readahead` size until you see no more improvement in transfer rates. The following example shows a method for testing timings on disk reads.

```
# timex dd if=/sam/myfile of=/dev/null bs=256k
```

You should test various `readahead` sizes for your environment. The `readahead` parameter should be set to a size that increases the I/O performance for paged I/O but is not so large as to hurt performance. You should also consider the amount of memory and number of concurrent streams. . Setting the `readahead` value multiplied by the number of streams to a value that is greater than memory can cause page thrashing.

The default `readahead` value is 1024 kilobytes. This value favors large-block, sequential I/O. For short-block, random I/O applications, set `readahead` to the typical request size. Database applications do their own read-ahead, so for these applications, set `readahead` to 0.

The `readahead` setting can be enabled from a mount option, from within the `samfs . cmd` file, from within the `/etc/vfstab` file, or from a command within the `samu` utility. For information about enabling this setting from a mount option, see the description of the `-o readahead=n` option on the `mount_samfs(1M)` man page. For information about enabling this setting from the `samfs . cmd` file, see “`samfs.cmd(4)`” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*. For information about enabling this setting from within `samu(1M)`, see “`samu(1M)`” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

7 Set the stripe width.

The `-o stripe=n` option with the `mount` command specifies the stripe width for the file system. The stripe width is based on the disk allocation unit (DAU) size. The *n* argument specifies that *n* x DAU bytes are written to one device before writing switches to the next device. The DAU size is set when the file system is initialized by the `sammkfs -a` command.

If `-o stripe=0` is set, files are allocated to file system devices using the round-robin allocation method. With this method, each file is completely allocated on one device until that device is full. Round-robin is the preferred setting for a multistream environment. If `-o stripe=n` is set to an integer greater than 0, files are allocated to file system devices using the stripe method. To determine the appropriate `-o stripe=n` setting, try varying the setting and taking performance readings. Striping is the preferred setting for turnkey applications with a required bandwidth.

You can also set the stripe width from the `/etc/vfstab` file or from the `samfs.cmd` file.

For more information about the `mount` command, see “[mount_samfs\(1M\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*. For more information about the `samfs.cmd` file, see “[samfs.cmd\(4\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

Enabling Qwrite Capability

By default, the Sun QFS file systems disable simultaneous reads and writes to the same file. This mode is defined by the UNIX vnode interface standard, which gives exclusive access to only one write while other writers and readers must wait. Qwrite enables simultaneous reads and writes to the same file from different threads.

The Qwrite feature can be used in database applications to enable multiple simultaneous transactions to the same file. Database applications typically manage large files and issue simultaneous reads and writes to the same file. Each system call to a file acquires and releases a read/write lock inside the kernel. This lock prevents overlapped or simultaneous operations to the same file. If the application itself implements file-locking mechanisms, the kernel-locking mechanism impedes performance by unnecessarily serializing I/O.

Qwrite can be enabled in the `/etc/vfstab` file, in the `samfs.cmd` file, and as a mount option. The `-o qwrite` option with the `mount` command bypasses the file system locking mechanisms except for applications accessing the file system through NFS and lets the application control data access. If `qwrite` is specified, the file system enables simultaneous reads and writes to the same file from different threads. This option improves I/O performance by queuing multiple requests at the drive level.

The following example uses the `mount` command to enable Qwrite on a database file system:

```
# mount -F samfs -o qwrite /db
```

For more information about this feature, see the description of the `qwrite` directive on the `samfs.cmd(4)` man page or the `-o qwrite` option on the `mount_samfs(1M)` man page.

Setting the Write Throttle

The `-o wr_throttle=n` option limits the number of outstanding write kilobytes for one file to *n*. By default, Sun QFS file systems set the `wr_throttle` to 16 megabytes.

If a file has *n* write kilobytes outstanding, the system suspends an application that attempts to write to that file until enough bytes have completed the I/O so that the application to be resumed.

If your site has thousands of streams, such as thousands of NFS-shared workstations accessing the file system, you can tune the `-o wr_throttle=n` option in order to avoid flushing excessive amounts of memory to disk at once. Generally, the number of streams multiplied by 1024 x the *n* argument to the `-o wr_throttle=n` option should be less than the total size of the host system's memory minus the memory needs of the Oracle Solaris OS, as shown in this formula:

$$\text{number-of-streams} \times n \times 1024 < \text{total-memory} - \text{Solaris-OS-memory-needs}$$

For turnkey applications, you might want to use a size larger than the default 16,384 kilobytes to keep more pages in memory.

Setting the Flush-Behind Rate

Two mount parameters control the flush-behind rate for pages written sequentially and for stage pages. The `flush_behind` and `stage_flush_behind` mount parameters are read from the `samfs.cmd` file, the `/etc/vfstab` file, or the mount command.

The `flush_behind=n` mount parameter sets the maximum flush-behind value. Modified pages that are being written sequentially are written to disk asynchronously to help the Solaris Volume Manager (SVM) layer keep pages clean. To enable this feature, set *n* to be an integer from 16 through 8192. By default, *n* is set to 0, which disables this feature. The *n* argument is specified in kilobyte units.

The `stage_flush_behind=n` mount parameter sets the maximum stage flush-behind value. Stage pages that are being staged are written to disk asynchronously to help the SVM layer keep pages clean. To enable this feature, set *n* to be an integer from 16 through 8192. By default, *n* is set to 0, which disables this feature. The *n* argument is specified in kilobyte units.

For more information about these mount parameters, see “[mount_samfs\(1M\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual* or “[samfs.cmd\(4\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

Tuning the Number of Inodes and the Inode Hash Table

The Sun QFS file system enables you to set the following two tunable parameters in the `/etc/system` file:

- `ninodes`
- `nhino`

To enable non-default settings for these parameters, edit the `/etc/system` file, and then reboot your system.

ninodes Parameter

The `ninodes` parameter specifies the maximum number of default inodes. The value for `ninodes` determines the number of in-core inodes that Sun QFS software keeps allocated to itself even when applications are not using many inodes.

The format for this parameter in the `/etc/system` file is as follows:

```
set samfs:ninodes = _value_
```

The range for *value* is from 16 through 2000000. The default value for `ninodes` is one of the following:

- A value that is equal to the `ncsize` setting. The `ncsize` parameter is an Oracle Solaris tuning parameter that specifies the number of entries in the directory name look-up cache (DNLC). For more information about `ncsize`, see the *Oracle Solaris Tunable Parameters Reference Manual*.
- `2000`. `ninodes` is set to `2000` if the `ncsize` setting is zero or out of range.

nhino Parameter

The `nhino` parameter specifies the size of the in-core inode hash table.

The format for this parameter in the `/etc/system` file is as follows:

```
set samfs:nhino = value
```

The range for *value* is 1 through 1048756. *value* must be a nonzero power of 2. The default value for `nhino` is one of the following:

- A value that is equal to the `ninodes` value divided by 8 and then, if necessary, rounded up to the nearest power of 2. For example, assume that the following line exists in `/etc/system`:

```
set samfs:ninodes 8000
```

For this example, if `nhino` is not set, the system assumes 1024, which is 8000 divided by 8 and then rounded up to the nearest power of 2 (210).

- `nhino` is set to 512 if the `ninodes` setting is out of range.

When to Set the `ninodes` and `nhino` Parameters

When searching for an inode by number, a Sun QFS file system searches its cache of in-core inodes. To speed this process, the file system maintains a hash table to decrease the number of inodes it must check.

A larger hash table reduces the number of comparisons and searches, at a modest cost in memory usage. If the `nhino` value is too large, the system is slower when undertaking operations that sweep through the entire inode list (inode syncs and unmounts). For sites that manipulate large numbers of files and sites that do extensive amounts of NFS I/O, setting these parameter values to larger than the defaults can be advantageous.

If your site has file systems that contain only a small number of files, consider making these numbers smaller than the defaults. For example, this setting might improve performance if you have a file system into which you write large single-file `tar(1)` files to back up other file systems.

Using QFS File Systems with SANergy (SAN-QFS)

This chapter provides information about using Sun QFS file systems with SANergy (SAN-QFS).

Note – The SANergy software is not supported on x64 hardware platforms.

Using the SAN-QFS File System in a Heterogeneous Computing Environment

The SAN-QFS file system enables multiple hosts to access the data stored in a Sun QFS system at full disk speeds. This capability can be especially useful for database, data streaming, web page services, or any application that demands high-performance, shared-disk access in a heterogeneous environment.

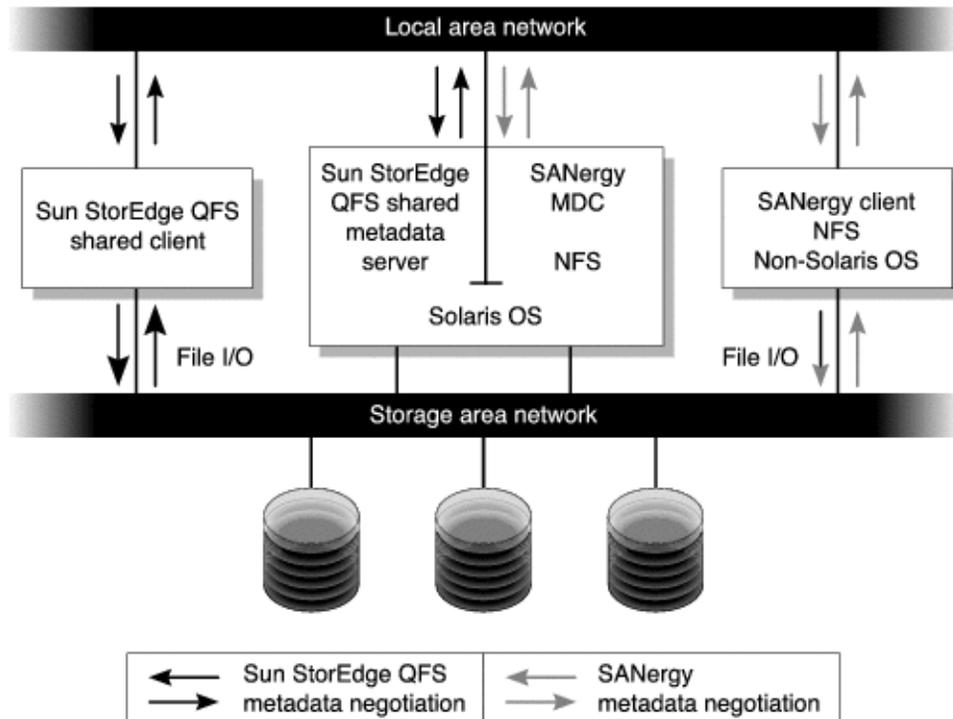
You can use the SAN-QFS file system in conjunction with fibre-attached devices in a storage area network (SAN). The SAN-QFS file system enables high-speed access to data through Sun QFS software and software such as Tivoli SANergy file-sharing software. To use the SAN-QFS file system, you must have both the SANergy version (2.2.4 or later) and the Sun QFS software. For information about the levels of Sun QFS and SANergy software that are supported, contact your sales representative.

Note – In environments that include the Oracle Solaris OS and supported Linux operating systems, use the Sun QFS shared file system, not the SAN-QFS file system, on the Oracle Solaris hosts. For information about the Sun QFS shared file system, see [Chapter 5, “Configuring a Shared File System.”](#)

[Figure 11–1](#) depicts a SAN-QFS file system that uses both the Sun QFS software and the SANergy software. The clients and the metadata controller (MDC) system manage metadata across the local area network (LAN). The clients perform I/O directly to and from the storage devices.

Note that all clients running only the Oracle Solaris OS are hosting the Sun QFS software, and that all heterogeneous clients running an OS other than Oracle Solaris are hosting the SANergy software and the NFS software. The SAN-QFS file system's metadata server hosts both the Sun QFS and the SANergy software. This server acts both as the metadata server for the file system and as the SANergy MDC.

FIGURE 11-1 SAN-QFS File System Using Sun QFS Software and SANergy Software



SAN-QFS Shared File System and Sun QFS Shared File System Comparison

The SAN-QFS shared file system and the Sun QFS shared file system have the following similarities:

- Both can stage files.
- Both are useful in data capture environments in which the primary file system host should not be responsible for writing the data.
- Both are advantageous in environments where there is contention for writing files.

The following table describes differences between the file systems.

TABLE 11-1 SAN-QFS Shared File System Versus Sun QFS Shared File System

SAN-QFS File System	Sun QFS Shared File System
Uses NFS protocol for metadata	Uses natural metadata
Preferred in heterogeneous computing environments (that is, when not all hosts are Sun systems)	Preferred in homogeneous Oracle Solaris OS environments
Useful in environments where multiple, heterogeneous hosts must be able to write data	Preferred when multiple hosts must write to the same file at the same time

The SANergy software does not enforce block quotas. Therefore, you can exceed a block quota when writing a file with the SANergy software. For more information on quotas, see [“Enabling Quotas” on page 97](#).

The SANergy software uses the NFS software for metadata operations, which means that the NFS close-to-open consistency model is used for file data and attributes. File data and attributes among SANergy clients do not support the POSIX coherency model for open files.

Enabling the SAN-QFS File System

The following procedures describe how to enable the SAN-QFS file system. Perform these procedures in the order in which they are presented.

Before You Begin

Before you enable the SAN-QFS file system, keep the following configuration considerations in mind and plan accordingly:

- Disks configured for use in a SAN-QFS file system cannot be under the control of a volume manager.
- For the Sun QFS metadata server to be enabled or relocated in a SAN-QFS environment, the new metadata server system must be configured as a SANergy metadata controller.
- A SAN-QFS file system does not recognize segmented files. Unexpected behavior can occur if segmented files are used within the SAN-QFS environment.
- Devices that are classified as `ms` or `md` devices in the Sun QFS `mc f` file are not supported in a SAN-QFS file system.

Note – This documentation assumes that your non-Solaris clients are hosting SANergy software and NFS software for file system sharing. The text and examples in this document reflect this configuration. If your non-Solaris clients host the Samba software instead of the NFS software, see your Samba documentation.

▼ How to Enable the SAN-QFS File System on the Metadata Controller

When you use the SAN-QFS file system, one host system in your environment acts as the SANergy metadata controller (MDC). This system is the host system upon which the Sun QFS file system resides.

1 Log in to the host upon which the Sun QFS file system resides and become superuser.

2 Verify that the Sun QFS file system is tested and fully operational.

3 Install and configure the SANergy software.

For instructions, see your SANergy documentation.

4 Verify the SANergy software release level:

```
# pkginfo -l SANergy
```

5 Ensure that the file system is mounted.

Use the `mount` command either to verify the mount or to mount the file system.

6 Enable NFS access to client hosts.

```
MDC# share -F nfs -d qfs-file-system-name /mount-point
```

For *qfs-file-system-name*, specify the name of your Sun QFS file system, for example, `qfs1`. For more information about the `share` command, see the `share(1M)` or `share_nfs(1M)` man page.

For *mount-point*, specify the mount point of *qfs-file-system-name*.

7 If you are connecting to Microsoft Windows clients, configure Samba, rather than NFS, to provide security and namespace features.

Add the `SANERGY_SMBPATH` environment variable in the `/etc/init.d/sanergy` file and point it to the location of the Samba configuration file. For example, if your Samba configuration file is named `/etc/swf/smb.conf`, add the following lines to the beginning of your `/etc/init.d/sanergy` file:

```
SANERGY_SMBPATH=/etc/swf/smb.confexport SANERGY_SMBPATH
```

- 8 (Optional) Edit the file system table (`/etc/dfs/dfstab`) on the MDC to enable access at boot time.

Perform this step if you want to automatically enable this access at boot time.

▼ How to Enable the SAN-QFS File System on the Clients

After you have enabled the file system on the MDC, you are ready to enable it on the client hosts. The SAN-QFS file system supports several client hosts including IRIX, Microsoft Windows, AIX, and Linux hosts. For information about the specific clients supported, see your Oracle sales representative.

Every client has different operational characteristics. This procedure uses general terms to describe the actions you must take to enable the SAN-QFS file system on the clients. For information specific to your clients, see the documentation provided with your client hosts.

- 1 Log in to each of the client hosts.
- 2 Edit the file system defaults table on each client and add the file system.

For example, on an Oracle Solaris OS, edit the `/etc/vfstab` file on each client and add the name of your Sun QFS file system, as follows:

```
server:/qfs1 - /qfs1 nfs - yes noac,hard,intr,timeo=1000
```

On other operating system platforms, the file system defaults table might reside in a file other than `/etc/vfstab`. For example, on Linux systems, this file is `/etc/fstab`.

For more information about editing the `/etc/vfstab` file, see [Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide](#). For information about required or suggested NFS client mount options, see your SANergy documentation.

▼ How to Install the SANergy Software on the Clients

After enabling the file system on the client hosts, you are ready to install the SANergy software on the clients. The following procedure describes the SANergy installation process in general terms.

- 1 Install and configure the SANergy software.
For instructions, see your SANergy documentation.
- 2 Use the `mount` command to NFS mount the file system.

For example:

```
# mount host:/mount-point /local-mount-point
```

For `host`, specify the MDC.

For *mount-point*, specify the mount point of the Sun QFS file system on the MDC.

For *local-mount-point*, specify the mount point on the SANergy client.

3 Fuse the software.

```
# fuse |mount-point
```

For *mount-point*, specify the mount point on the SANergy client.

Unmounting the SAN-QFS File System

The following procedures describe how to unmount a SAN-QFS file system that is using the SANergy software. Perform these procedures in the order in which they are presented.

▼ How to Unmount the SAN-QFS File System on the SANergy Clients

Follow these steps for each client host on which you want to unmount the SAN-QFS file system.

1 Log in to the client host and become superuser.

2 Unfuse the file system from the software.

```
# unfuse |mount-point
```

For *mount-point*, specify the mount point on the SANergy client.

3 Unmount the file system from NFS.

```
# umount host:/mount-point /local-mount-point
```

For *host*, specify the MDC.

For *mount-point*, specify the mount point of the Sun QFS file system on the MDC.

For *local-mount-point*, specify the mount point on the SANergy client.

▼ How to Unmount the SAN-QFS File System on the Metadata Controller

1 Log in to the MDC system and become superuser.

2 Disable NFS access to client hosts.

```
MDC# unshare qfs-file-system-name /mount-point
```

For *qfs-file-system-name*, specify the name of your Sun QFS file system, such as `qfs1`.

For *mount-point*, specify the mount point of *qfs-file-system-name*.

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

▼ How to Unmount the SAN-QFS File System on the Sun QFS Clients

Follow these steps on each participating client host.

- 1 Log in to a Sun QFS client host and become superuser.
- 2 Unmount the file system.

For example:

```
# umount /qfs1
```

▼ How to Unmount the SAN-QFS File System on the Sun QFS Server

- 1 Log in to the host system upon which the Sun QFS file system resides and become superuser.
- 2 Unmount the file system.

Troubleshooting: Unmounting a SAN-QFS File System With SANergy File Holds

SANergy software issues holds on Sun QFS files to reserve them temporarily for accelerated access. If SANergy crashes when holds are in effect, you will not be able to unmount the file system. If you are unable to unmount a SAN-QFS file system, examine the `/var/adm/messages` file and look for console messages that describe outstanding SANergy holds.

Whenever possible, allow the SANergy file-sharing function to clean up its holds. In an emergency or in case of a SANergy file-sharing system failure, use the following procedure to avoid a reboot.

▼ How to Unmount a File System in the Presence of SANergy File Holds

1 Use the `unshare` command to disable NFS access.

2 Use the `samunhold` command to release the SANergy file system holds.

For more information about this command, see “`samunhold(1M)`” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

3 Use the `umount` command to unmount the file system.

Using `samgrowfs` to Expand SAN-QFS File Systems

You can use the `samgrowfs` command to increase the size of a SAN-QFS file system. To perform this task, follow the procedures described in “[Adding Disk Cache to a File System](#)” on page 90.



Caution – When using this procedure, be aware that the line-by-line device order in the `mcf` file must match the order of the devices listed in the file system's superblock.

When the `samgrowfs` command is issued, the devices that were already in the `mcf` file keep their positions in the superblock. New devices are written to subsequent entries in the order in which they are encountered.

If this new order does not match the order in the superblock, the SAN-QFS file system cannot be fused.

Mount Options in a Shared File System

The Sun QFS shared file system can be mounted with several mount options. This chapter describes some of these options within the context of their roles.

Shared File System Mount Options

You can specify most mount options by using the mount command, by entering them in the `/etc/vfstab` file, or by entering them in the `samfs .cmd` file. For example, the following `/etc/vfstab` file includes mount options for a shared file system:

```
sharefs1 - /sfs samfs - no shared,mh_write
```

You can change some mount options dynamically by using the `samu(1M)` operator utility. For more information about these options, see [Chapter 13, “Using the samu Operator Utility.”](#)

For more information about any of these mount options, see “`mount_samfs(1M)`” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual* or see the cross-references mentioned in their descriptions.

Mounting in the Background: (bg Option)

The `bg` mount option specifies that if the first mount operation fails, subsequent attempts at mounting should occur in the background. By default, `bg` is not in effect, and mount attempts continue in the foreground.

Reattempting a File System Mount: (retry Option)

The `retry` mount option specifies the number of times that the system should attempt to mount a file system. The default is 10000.

Declaring a Sun QFS Shared File System: (shared Option)

The `shared` mount option declares a file system to be a Sun QFS shared file system. This option must be specified in the `/etc/vfstab` file in order for the file system to be mounted as a Sun QFS shared file system. The presence of this option in a `samfs.cmd` file or on the `mount` command does not cause an error condition, but it does not mount the file system as a shared file system.

Tuning Allocation Sizes: (minallocsz and maxallocsz Options)

The `minallocsz` and `maxallocsz` options to the `mount` command specify an amount of space, in kilobytes. These options set the minimum block allocation size. If a file is growing, the metadata server allocates blocks when an append lease is granted. Use `-o minallocsz=n` to specify the initial size of this allocation. The metadata server can increase the size of the block allocation depending on the application's access patterns up to but not exceeding the `-o maxallocsz=n` setting.

You can specify these mount options on the `mount` command line, in the `/etc/vfstab` file, or in the `samfs.cmd` file.

Using Leases in a Sun QFS Shared File System: (rdlease, wrlease, and aplease Options)

A *lease* grants a shared host permission to perform an operation on a file for as long as the lease is valid. The metadata server issues leases to each shared host, including itself. The leases are renewed as necessary to permit continued file operations. The possible file operations are as follows:

- A *read lease* enables existing file data to be read.
- A *write lease* enables existing file data to be overwritten.
- An *append lease* enables a file's size to be extended and enables newly allocated blocks to be written.

A shared host can continue to update leases for as long as necessary. The lease is transparent to the end user. The following table shows the mount options that enable you to specify the duration of each lease type.

TABLE 12-1 Lease-Related mount(1M) Options

Option	Action
-o rdlease= <i>n</i>	Specifies the maximum amount of time, in seconds, for the read lease.
-o wrlease= <i>n</i>	Specifies the maximum amount of time, in seconds, for the write lease.
-o aplease= <i>n</i>	Specifies the maximum amount of time, in seconds, for the append lease.

All three leases enable you to specify an *n* such that $15 \leq n \leq 600$. The default time for each lease is 30 seconds. A file cannot be truncated if a lease is in effect. For more information about setting these leases, see “[mount_samfs\(1M\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

If you change the metadata server because the current metadata server is down, you must add the lease time to the changeover time because all leases must expire before an alternate metadata server can assume control.

Setting a short lease time causes more traffic between the client hosts and the metadata server because the lease must be renewed after it has expired.

Enabling Multiple Host Reads and Writes: (mh_write Option)

By default, in a Sun QFS shared file system, multiple hosts can read the same file at the same time. If no host is writing to that file, I/O can be paged on all hosts. Only one host can append or write to a file at any one time.

The `mh_write` option controls write access to the same file from multiple hosts. If `mh_write` is specified as a mount option on the metadata server host, the Sun QFS shared file system enables simultaneous reads and writes to the same file from multiple hosts. If `mh_write` is not specified on the metadata server host, only one host can write to a file at any one time.

By default, `mh_write` is disabled, and only one host has write access to a file at any one time. The length of that time period is determined by the duration of the `wrlease` mount option. If the Sun QFS shared file system is mounted on the metadata server with the `mh_write` option enabled, simultaneous reads and writes to the same file can occur from multiple hosts.

The following table describes how file access from multiple hosts is affected depending on whether the `mh_write` option is enabled on the metadata server.

TABLE 12-2 File Access Based on the `mh_write` Option

<code>mh_write</code> Not Enabled on the Metadata Server	<code>mh_write</code> Enabled on the Metadata Server
Multiple reader hosts allowed. Can use paged I/O.	Multiple reader hosts allowed. Can use paged I/O.
Only one writer host is allowed. Can use paged I/O. All other hosts wait.	Multiple reader and/or writer hosts allowed. If any writer hosts exist, all I/O is direct.
Only one append host. All other hosts wait.	Only one append host is allowed. All other hosts can read or write. If any writer hosts exist, all I/O is direct.

The `mh_write` option does not change locking behavior. File locks behave the same regardless of whether `mh_write` is in effect. The `mh_write` option's effect is as follows:

- When `mh_write` is in effect, all hosts can read from and write to the same file simultaneously.
- When `mh_write` is not in effect, only one host can write to a given file during a given time interval, and no hosts can read from the file during that time interval.

Sun QFS shared file system maintains consistency between hosts. The first time that a host executes a read or write system call, it gets a *lease*, which enables it to read or write the file for some period of time. The existence of that lease prevents other hosts without `mh_write` from accessing the file. In particular, the lease can last longer than the duration of the system call that caused its acquisition.

When `mh_write` is not in effect, the Sun QFS shared file system should provide near-POSIX behavior for data reads and writes. For metadata, however, access time changes might not be seen immediately on other hosts. Changes to a file are pushed to disk at the end of a write lease. When a read lease is acquired, the system invalidates any stale cache pages so that the newly written data can be seen.

When `mh_write` is in effect, behavior might be less consistent. When there are simultaneous readers and writers, the Sun QFS shared file system switches all hosts accessing the file into direct I/O mode. Therefore, page-aligned I/O should be visible immediately to other hosts. However, non-page-aligned I/O can result in stale data being visible, or even written to the file, because the normal lease mechanism preventing such occurrences has been disabled.

You should specify the `mh_write` option only when multiple hosts need to write to the same file simultaneously and when applications perform page-aligned I/O. In other cases, data inconsistency could occur because even using `flock()` (which works with `mh_write`) to coordinate between hosts does not guarantee consistency.

For more information about `mh_write`, see “[mount_samfs\(1M\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

Setting the Minimum Number of Concurrent Threads: (min_pool Option)

The `min_pool` mount option sets the minimum number of concurrent threads for the Sun QFS shared file system. The default setting is `min_pool=64` on Oracle Solaris systems. This setting means that at least 64 active threads will be in the thread pool on Oracle Solaris. You can adjust the `min_pool` setting to any value between 8 and 2048, depending on the Sun QFS shared file system's activity.

The `min_pool` mount option must be set in the `samfs.cmd` file. It will be ignored if set in the `/etc/vfstab` file or at the command line.

Note – The `min_pool` mount option replaces the previous `nstreams` mount option. In version 5.0 of the software, the `nstreams` option is completely removed.

Retaining Cached Attributes: (meta_timeo Option)

The `meta_timeo` mount option determines how long the system waits between checks on the metadata information. By default, the system refreshes metadata information every three seconds. For example, an `ls` command entered in a Sun QFS shared file system with several newly created files might not return information about all the files until three seconds have passed. The syntax for the option is `meta_timeo=n` specifies a value such that $0 \leq n \leq 60$.

Specifying Striped Allocation: (stripe Option)

By default, data files in the Sun QFS shared file system are allocated using the round-robin file allocation method. To specify that file data be striped across disks, you can specify the `stripe` mount option on the metadata host and all potential metadata hosts. Note that by default, unshared file systems allocate file data using the striped method.

In a round-robin allocation, files are created in a round-robin fashion on each slice or striped group. The maximum performance for one file will be the speed of a slice or striped group. For more information about file allocation methods, see [“Sun QFS File Systems Design Basics” on page 19](#).

Specifying the Frequency With Which Metadata Is Written: (sync_meta Option)

You can set the `sync_meta` option to `sync_meta=1` or `sync_meta=0`.

The default setting is `sync_meta=1`, which means that a Sun QFS shared file system writes file metadata to disk every time the metadata changes. This setting slows data performance but ensures data consistency. This setting must be in effect if you want to change the metadata server.

If you set `sync_meta=0`, the Sun QFS shared file system writes the metadata to a buffer before writing it to disk. This delayed write delivers higher performance but decreases data consistency after an unscheduled machine interruption.

Enabling WORM Functionality: (worm_capable and def_retention Options)

If you are using the optional WORM package, the `worm_capable` mount option enables the file system to support WORM files. The `def_retention` mount option sets the default retention time using the format `def_retention=MyNdOhPm`.

In this format, *M*, *N*, *O*, and *P* are non-negative integers and *y*, *d*, *h*, and *m* stand for years, days, hours, and minutes, respectively. Any combination of these units can be used. For example, `1y5d4h3m` indicates 1 year, 5 days, 4 hours, and 3 minutes; `30d8h` indicates 30 days and 8 hours; and `300m` indicates 300 minutes. This format is backward compatible with the formula in previous software versions, in which the retention period was specified in minutes.

See [Chapter 9, “Configuring WORM-FS File Systems,”](#) for more information about the WORM functionality.

Using the samu Operator Utility

This chapter describes how to use the `samu` command to control the devices configured within your environment. Many `samu` displays are useful only for sites using the storage and archive management mechanism. If you are using `samu` in a Sun QFS-only environment, these displays do not apply to you.

You can also use the `samcmd` command to perform many of the same operations. For more information, see “`samcmd(1M)`” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

samu Operator Utility at a Glance

Option	Description	For More Information
a	Displays archiver status	“(a) - Archiver Status Display” on page 179
c	Displays device configuration	“(c) - Device Configuration Display” on page 180
C	Displays memory information	“(C) - Memory Display” on page 182
d	Displays traced events	“(d) - Daemon Trace Controls Display” on page 183
D	Displays the disk volume dictionary	“(D) - Disk Volume Dictionary” on page 184
f	Displays the components of a Sun QFS file systems	“(f) - File Systems Display” on page 185
F	Displays the label on an optical disk	“(F) - Optical Disc Label Display” on page 187

Option	Description	For More Information
h	Displays a summary of the available samu(1M) displays	“(h) - Help Display” on page 187
I	Displays the content of inodes	“(I) - Inode Display” on page 188
J	Displays the shared memory segment for the preview queue	“(J) - Preview Shared Memory Display” on page 189
K	Displays kernel statistics	“(K) - Kernel Statistics Display” on page 190
l	Displays usage information for the file system	“(l) - Usage Display” on page 191
L	Displays the location of shared memory tables and system defaults in shared memory	“(L) - Shared Memory Tables” on page 191
m	Displays the status of mass storage mounted file systems and member drives	“(m) - Mass Storage Status Display” on page 192
N	Displays information about the file system	“(N) - File System Parameters Display” on page 195
o	Displays the status of optical disc drives	“(o) - Optical Disc Status Display” on page 196
p	Displays pending load requests for removable media	“(p) - Removable Media Load Requests Display” on page 197
P	Displays the services registered with the Sun QFS single-port multiplexer	“(P) - Active Services Display” on page 199
r	Displays the activity of removable media devices	“(r) - Removable Media Status Display” on page 200
R	Displays information about and the status of Sun SAM-Remote configurations	“(R) - Sun SAM-Remote Information Display” on page 201
s	Displays the status for configured devices	“(s) - Device Status Display” on page 201
S	Displays raw device data	“(S) - Sector Data Display” on page 202
t	Displays the status of all configured tape drives	“(t) - Tape Drive Status Display” on page 203
T	Displays the SCSI status of a SCSI device	“(T) - SCSI Sense Data Display” on page 204

Option	Description	For More Information
u	Displays the list of files in the staging queue	“(u) - Staging Queue Display” on page 205
U	Displays the device table in a human-readable form	“(U) - Device Table Display” on page 206
v	Displays the location and VSN of all cataloged disks or tapes	“(v) - Automated Library Catalog Display” on page 207
w	Displays pending stage requests with unloaded volumes	“(w) - Pending Stage Queue” on page 210
down	Terminates operation on device	“Device Commands” on page 216
idle	Restricts access to device by preventing new connections	“Device Commands” on page 216
off	Logically turns off device	“Device Commands” on page 216
on	Logically turns on device	“Device Commands” on page 216
unavail	Makes device unavailable for use with the file system	“Device Commands” on page 216
unload	Unloads the mounted media for the specified removable media device	“Device Commands” on page 216
nalloc	Prohibits any future allocation to the device	“Device Commands” on page 216
alloc	Re-enables allocation to the device	“Device Commands” on page 216
flush_behind	Controls whether modified pages are written to disk asynchronously	“The flush_behind Command” on page 216
force_nfs_async noforce_nfs_async	Controls whether the file system caches NFS data written to the server	“force_nfs_async and noforce_nfs_async Commands” on page 217
sw_raid nosw_raid	Controls whether the file system aligns the writebehind buffer.	“sw_raid and nosw_raid Commands” on page 218
readahead	Sets the maximum number of bytes that can be read ahead by the file system	“readahead Command” on page 217
writebehind	Sets the maximum number of bytes that can be written behind by a file system.	“writebehind Command” on page 218

Option	Description	For More Information
<code>wr_throttle</code>	Sets the number of outstanding write kilobytes for one file.	“wr_throttle Command” on page 218
<code>dio_rd_form_min</code>	Sets the minimum number of blocks for read operations for well-aligned I/O	“dio_rd_form_min and dio_wr_form_min Commands” on page 219
<code>dio_wr_form_min</code>	Sets the minimum number of blocks for write operations for well-aligned I/O	“dio_rd_form_min and dio_wr_form_min Commands” on page 219
<code>dio_rd_ill_min</code>	Sets the minimum number of blocks for read operations for misaligned I/O	“dio_rd_ill_min and dio_wr_ill_min Commands” on page 220
<code>dio_wr_ill_min</code>	Sets the minimum number of blocks for write operations for misaligned I/O	“dio_rd_ill_min and dio_wr_ill_min Commands” on page 220
<code>dio_rd_consec</code>	Sets the number of consecutive read operations that can occur without regard to a buffer size	“dio_rd_consec and dio_wr_consec Commands” on page 220
<code>dio_wr_consec</code>	Sets the number of consecutive write operations that can occur without regard to a buffer size	“dio_rd_consec and dio_wr_consec Commands” on page 220
<code>dio_szero</code> <code>nodio_szero</code>	Controls whether uninitialized areas of sparse files written with direct I/O are zeroed when the area is accessed.	“dio_szero and nodio_szero Commands” on page 220
<code>forcedirectio</code> <code>noforcedirectio</code>	Controls whether direct I/O or buffered I/O is used as the default mode.	“forcedirectio and noforcedirectio Commands” on page 221
<code>meta_timeo</code>	Sets the time limit for the Sun QFS shared file system metadata cache	“meta_timeo Command” on page 221
<code>mh_write</code> <code>nomh_write</code>	Controls whether to allow multihost read and write operations	“mh_write and nomh_write Commands” on page 221
<code>minallocsz</code>	Sets the minimum number of allocated blocks	“minallocsz and maxallocsz Commands” on page 222
<code>maxallocsz</code>	Sets the maximum number of allocated blocks	“minallocsz and maxallocsz Commands” on page 222
<code>rdlease</code>	Sets the amount of time for read leases	“rdlease, wrlease, and aplease Commands” on page 222

Option	Description	For More Information
wrlease	Sets the amount of time for write leases	“rdlease, wrlease, and aplease Commands” on page 222
aplease	Sets the amount of time for append leases	“rdlease, wrlease, and aplease Commands” on page 222
abr noabr	Controls the application binary recovery (ABR) mount option.	“abr and noabr Commands” on page 222
dmr nodmr	Controls the direct mirror reads (DMR) mount option.	“dmr and nodmr Commands” on page 223
invalid	Sets the number of seconds that the file system holds cached attributes	“invalid interval Command” on page 223
mm_stripe	Sets the number of disk allocation units for the metadata stripe width	“mm_stripe Command” on page 223
qwrite noqwrite	Controls whether read and write operations to the same file are performed simultaneously from different threads	“qwrite and noqwrite Commands” on page 223
refresh_at_eof norefresh_at_eof	Controls whether Sun QFS hosts update file sizes	“refresh_at_eof and norefresh_at_eof Commands” on page 224
suid nosuid	Controls whether running programs can change their owner IDs	“suid and nosuid Commands” on page 224
stripe	Sets the stripe width for the file system to the specified number of disk allocation units	“stripe Command” on page 224
sync_meta	Controls whether metadata is written to disk immediately	“sync_meta Command” on page 225
trace notrace	Controls whether a file system uses the trace feature	“trace and notrace Commands” on page 225
clear	Removes the specified volume from the Removable Media Mount Requests display	“clear vsn Command” on page 226
devlog	Specifies one or more events to be logged	“devlog Command” on page 226
diskvols	Controls the flags in the disk volume dictionary	“diskvols flag Command” on page 226

Option	Description	For More Information
<code>dt trace</code>	Control the DTrace feature for one or more processes	“dt trace Commands” on page 227
<code>fs</code>	Sets the default file system	“fs Command” on page 227
<code>mount</code>	Specifies a Sun QFS file system	“mount Command” on page 228
<code>open</code>	Enables access to a disk device	“open Command” on page 228
<code>read</code>	Reads the specified sector from an open disk device	“read Command” on page 228
<code>refresh</code>	Sets the amount of time between samu(1M) screen refreshes	“refresh Command” on page 228
<code>snap</code>	Sends a copy of an operator display to file	“snap Command” on page 228
<code>!</code>	Enables you to run a shell command within the samu(1M) operator utility	“! shell-command Command” on page 228

Overview of the samu Operator Utility

The samu operator utility requires a display terminal that displays a minimum of 24 lines by 80 characters wide. The utility includes the following features:

- Displays that enable you to monitor Sun QFS and SAM-QFS devices and file system activity
- Commands that enable you to select displays, set display options, control access to and the activity of devices, and take snapshots of display windows
- Commands that enable you to tune a running file system

The display windows shown in this section are representative examples. The format and amount of information displayed on your terminal can be different depending on your terminal model and the devices configured in your environment.

The following sections describe how to start and stop samu, interact with the utility, access the help windows, and view operator displays.

▼ How to Start the samu Utility

- **To start the operator utility, type the samu command from the command line.**

```
# samu
```

The system starts samu and shows the help display, which is the default initial display. To view a different samu display, follow the steps in [“How to Display a samu Screen” on page 177](#).

To change the default initial display, see “samu(1M)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

Note – samu, like the vi editor, is based on the curses(3CURSES) routine. If you have trouble invoking samu, make sure that your terminal type is defined correctly.

▼ How to Display a samu Screen

The samu command accepts options on its command line for displaying different samu screens.

1 Type a colon (:) character.

The following command line appears in the lower left corner:

Command:

2 Type the letter that corresponds to the display you want to view and press the Return key.

For example, to view the Automated Library Catalog Display, type a v character and press Return.

For a complete list of the displays and the letters that correspond to the displays, type h for the Help Display.

▼ How to Stop samu

● To exit samu, type one of the following:

- q
- :q

The samu operator utility exits and returns you to the command shell.

Interacting With samu

The samu utility is similar the UNIX vi editor with respect to paging forward or backward, entering commands, refreshing the display, and quitting the utility.

For each operator display, the description includes the control key sequences you use to navigate in that display. The samu(1M) man page summarizes the control key sequences.

The last line of the display window shows any error messages. If a command error occurs, automatic display refreshing halts until the next operator action.

Identifying a Device

Each device included in the Sun QFS environment is assigned an equipment number (for example, 10) in the `mc f` file. Many `samu` commands require you to identify a specific device using its equipment number. To see a list of the devices and their equipment numbers, access the (c) - Device Configuration Display

Accessing Online Help

When you start `samu`, the default display is the first page of the online Help. The Help provides a summary of the `samu` displays.

To access the Help display at any time, type `:h`. To navigate in the online Help, press `Ctrl-F` to move to the next page, or `Ctrl-B` to move to the previous page. To return to the help display at any time, press the `h` key.

Operator Displays

To view one of the operator displays, press its corresponding character key. The lowercase characters `a` through `w` display operational information.

Note – The uppercase `samu(1M)` displays (`C`, `D`, `F`, `I`, `J`, `K`, `L`, `M`, `N`, `P`, `R`, `S`, `T`, and `U`) are designed to be used at a customer site only with the assistance of a member of the technical support staff.

This topic does not describe these uppercase displays in detail.

When an operator display overflows the available screen area, the word `more` appears on the bottom of the display indicating additional information, as shown in the following example. Press `Ctrl-F` to page forward and see more content.

EXAMPLE 13-1 `samu` Screen Indicating More Text Can Be Obtained

```
xb54 54  exb8505  pt03  0  yes  2  0  on
lt55 55  dlt2000  pt02  1  yes  4  0  on  ml65
hp56 56  hpc1716  pt01  1  yes  3  0  on  hp70
hp57 57  hpc1716  pt01  1  yes  4  0  on  hp70
more
```

When `samu` prompts you to provide a device, enter its associated equipment number. The configuration display (c) shows equipment numbers for all removable media devices. To control all displays, use the control keys listed for the display.

The following sections describe the operator displays in alphabetical order with examples. Descriptions of more complex displays are followed by a table describing the field in the display.

Note – If you are using only the Sun QFS software without the archiving features of SAM-QFS, not all operator displays will be available.

(a) - Archiver Status Display

The a display shows the archiver status. To display archiving details for a specific file system:

a *filesystem*

For *filesystem*, specify the name of a file system.

Navigation

The following table shows the control keys you can use in the a display.

Key	Function
Ctrl-B	Previous file system
Ctrl-F	Next file system
Ctrl-D	Page <i>arcopies</i> forward (bottom portion) — not available when viewing the details for a specified file system
Ctrl-U	Page <i>arcopies</i> backward (bottom portion) — not available when viewing the details for a specified file system

Sample Display

The following example shows the archiver status display.

EXAMPLE 13-2 samu a Display

```

Archiver Status
samu 5.0 07:44:02 August 8 2008
sam-archiverd: Waiting for resources
sam-arfind: samfs1 mounted at /sam1
Waiting until 2005-05-08 07:54:02 to scan .inodes
sam-arfind: samfs2 mounted at /sam2
Waiting until 2005-05-08 07:52:57 to scan .inodes
sam-arfind: qfs1 mounted at /qfs1
Waiting until 2005-05-08 07:44:33 to scan .inodes
sam-arfind: qfs2 mounted at /qfs2
Waiting until 2005-05-08 07:53:21 to scan .inodes
sam-arfind: qfs3 mounted at /qfs3
Waiting until 2005-05-08 07:44:11 to scan .inodes

```

EXAMPLE 13-2 samu aDisplay (Continued)

```

sam-arfind: qfs4 mounted at /qfs4
Waiting until 2005-05-08 07:53:35 to scan .inodes

sam-arfind: shareqfs1 mounted at /shareqfs1
Shared file system client. Cannot archive.

sam-arfind: shareqfs2 mounted at /shareqfs2
Shared file system client. Cannot archive.

sam-arcopy: qfs4.arset5.1.83 dt.DAT001
Waiting for volume dt.DAT001

```

Field Descriptions

The following table shows the fields in the detail display.

Field	Description
samfs1 mounted at	Mount point.
regular files	Number of regular files and their total size.
offline files	Number of offline files and their total size.
archdone files	Number of archdone files and size. Indicates that the archiver has completed processing and can perform no further processing. Files marked as archdone have been processed for archiving but have not necessarily been archived.
copy1	Number of files and total size for archive copy 1.
copy2	Number of files and total size for archive copy 2.
copy3	Number of files and total size for archive copy 3.
copy4	Number of files and total size for archive copy 4.
Directories	Number of directories and total size.
sleeping until	Indicates when archiver runs again.

(c) - Device Configuration Display

The c display shows your configuration's connectivity. It lists all device names and equipment numbers.

Navigation

The following table shows the control keys you can use in this display.

Key	Function
Ctrl-B	Page backward
Ctrl-D	Half-page forward
Ctrl-F	Page forward
Ctrl-U	Half-page backward

Sample Display

The following example shows the device configuration display.

EXAMPLE 13-3 samu c Display

```
Device configuration:          samu      5.0 07:48:11 Sept 8 2008
ty eq state  device_name      fs family_set
sk 100 on    /etc/opt/SUNWsamfs/dcstkconf 100 dcL700
tp 120 off   /dev/rmt/1cbn                 100 dcL700
sg 130 on    /dev/rmt/4cbn                 100 dcL700
sg 140 on    /dev/rmt/5cbn                 100 dcL700
tp 150 off   /dev/rmt/3cbn                 100 dcL700
hy 151 on    historian                    151
```

Field Descriptions

The following table shows the field descriptions for this display.

Field	Description
ty	Device type.
eq	Equipment number of the device.

Field	Description
state	<p>Current operating state of the device. Valid device states are as follows:</p> <ul style="list-style-type: none"> ▪ on The device is available for access. ▪ ro The device is available for read-only access. ▪ off The device is not available for access. ▪ down The device is available only for maintenance access. ▪ idle The device is not available for new connections. Operations in progress continue until completion. ▪ nalloc The nalloc flag has been set, which prohibits any future allocation to this device. For more information, see “Per-Logical Unit Number (LUN) Allocation Control” on page 38.
device_name	Path to the device.
fs	Family set equipment number.
family_set	Name of the storage family set or library to which the device belongs.

(C) - Memory Display

The C display shows the content of a specific memory address. To show the content at an address, type the address in hexadecimal.

To invoke this display:

c *hex-address*

For *hex-address*, specify the address of a memory location in hexadecimal. For example:

Command: C 0x1044a998

This display is designed for debugging. It is intended to be used only with the assistance of an Oracle support staff person.

Sample Display

The following example shows the memory display. The sample output has been truncated.

EXAMPLE 13-4 samu C Display

```

Memory      base: 0x1234567          samu 5.0 07:52:25 Sept 8 2008
00000000 80921000 137ff801 edd05e21 40853420 .....x.mP^!@.4
00000010 00a00420 018528b0 01a604e0 02840640 . . . .(0.&amp;.&rdquo;...@
00000020 02d030a1 a0853420 0080a0a0 100a6fff .P0! .4 . . .o.
00000030 f6921000 13c65e23 582d0000 0ba01020 v....F^#X-... .
00000040 00c45e20 48c608e0 2fd05e21 40920080 .D^ HF.&rdquo;/P^!@...
00000050 037ff801 fa941000 16c45e20 48a600a0 ..x.z...D^ H&amp;.;
00000060 80921000 137ff801 d5d05e21 40853420 .....x.UP^!@.4
00000070 00a00420 018528b0 01a604e0 02840640 . . . .(0.&amp;.&rdquo;...@
00000080 02d030a1 c0853420 0080a0a0 100a6fff .P0!@.4 . . .o.
00000090 f6921000 13c65e23 58a01020 00c45e20 v....F^#X . .D^
000000a0 48c608e0 2fd05e21 40920080 037ff801 HF.&rdquo;/P^!@.....x.
000000b0 e39405a2 00c45e20 48a600a0 80921000 c..".D^ H&amp;.; ....
000000c0 137ff801 bed05e21 40853420 00a00420 ..x.>P^!@.4 . .
000000d0 018528b0 01a604e0 02840640 02d030a1 ..(0.&amp;.&rdquo;...@.P0!
000000e0 e0853420 0080a0a0 100a6fff f6921000 &rdquo;.4 . . .o.v...
000000f0 13c65e23 58a01020 00c45e20 48c608e0 .F^#X . .D^ HF.&rdquo;

```

(d) - Daemon Trace Controls Display

The d display shows the events being traced as specified in the `defaults.conf` file. For more information about enabling trace files, see “[defaults.conf\(4\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

Sample Display

The following example shows trace file information. It includes information about the daemons being traced, the paths to the trace files, the events being traced, and information about the size and age of the trace files.

EXAMPLE 13-5 samu d Display

```

Daemon trace controls          samu 5.0 07:56:38 Sept 8 2008
sam-amld      /var/opt/SUNWsamfs/trace/sam-amld
              cust err fatal misc proc debug date
              size 0 age 0
sam-archiverd /var/opt/SUNWsamfs/trace/sam-archiverd
              cust err fatal misc proc debug date
              size 0 age 0
sam-catserverd /var/opt/SUNWsamfs/trace/sam-catserverd
              cust err fatal misc proc debug date
              size 0 age 0
sam-fsd       /var/opt/SUNWsamfs/trace/sam-fsd
              cust err fatal misc proc debug date
              size 0 age 0
sam-rftd      /var/opt/SUNWsamfs/trace/sam-rftd
              cust err fatal misc proc debug date
              size 0 age 0
sam-recycler  /var/opt/SUNWsamfs/trace/sam-recycler
              cust err fatal misc proc debug date
              size 0 age 0

```

EXAMPLE 13-5 samu d Display (Continued)

```

sam-sharefsd /var/opt/SUNWsamfs/trace/sam-sharefsd
             cust err fatal misc proc debug date
             size 0 age 0
sam-stagerd  /var/opt/SUNWsamfs/trace/sam-stagerd
             cust err fatal misc proc debug date
             size 0 age 0
sam-serverd  /var/opt/SUNWsamfs/trace/sam-serverd
             cust err fatal misc proc debug date
             size 0 age 0
sam-clientd  /var/opt/SUNWsamfs/trace/sam-clientd
             cust err fatal misc proc debug date
             size 0 age 0
sam-mgmt     /var/opt/SUNWsamfs/trace/sam-mgmt
             cust err fatal misc proc debug date
             size 0 age 0

```

(D) - Disk Volume Dictionary

The D display shows the disk volume dictionary, which keeps track of the disk media for disk archiving that has been defined in the `diskvols.conf` file. The dictionary contains information about each VSN, including the capacity, space remaining, and the status flags of the VSN. These flags include unavailable, read only, and bad media.

Sample Display

The following example shows the device configuration display.

EXAMPLE 13-6 samu D Display

```

Disk volume dictionary      samu      5.0 07:48:11 May 8 2008

header
version 460

volumes
magic 340322 version 9 nkeys 2 ndata 2
index  space          capacity      used        flags      volume
   0    12882411520    12887785472  10291200    -----  disk01
   1    6443827200     6443892736   70656      -----  disk02

clients
magic 340322 version 9 nkeys 1 ndata 1

```

Flags

The following table shows the flags for the D display.

Flag	Description
l	Volume is labeled; seqnum file has been created. This is set by the administrator to prevent the software from creating a new seqnum file.
r	Volume is defined on a remote host.
U	Volume is unavailable.
R	Volume is read only.
E	Media error, indicating the software detects a write error on the disk archive directory.

To set or clear a disk volume dictionary flag, use the `diskvols samu` command. For more information, see the `diskvols` command.

(f) - File Systems Display

The `f` display shows the components of your Sun QFS file systems.

Sample Display

The following example shows the file systems display.

EXAMPLE 13-7 samu fDisplay

```
File systems                                     samu 5.0 08:11:24 Sept 8 2008
```

ty	eq	state	device_name	status	high	low	mountpoint	server
ms	10	on	samfs1	m---2----	90%	70%	/sam1	
md	11	on	/dev/dsk/c5t8d0s3					
md	12	on	/dev/dsk/c5t8d0s4					
md	13	on	/dev/dsk/c5t8d0s5					
md	14	on	/dev/dsk/c5t8d0s6					
md	15	on	/dev/dsk/c5t8d0s7					
ms	20	on	samfs2	m---2----	90%	70%	/sam2	
md	21	on	/dev/dsk/c5t9d0s3					
md	22	on	/dev/dsk/c5t9d0s4					
md	23	on	/dev/dsk/c5t9d0s5					
md	24	on	/dev/dsk/c5t9d0s6					
md	25	on	/dev/dsk/c5t9d0s7					
ma	30	on	qfs1	m---2----	90%	70%	/qfs1	
mm	31	on	/dev/dsk/c5t10d0s0					
md	32	on	/dev/dsk/c5t10d0s1					
ma	40	on	qfs2	m---2----	90%	70%	/qfs2	
mm	41	on	/dev/dsk/c5t11d0s0					
md	42	on	/dev/dsk/c5t11d0s1					
ma	50	on	qfs3	m---2---r-	90%	70%	/qfs3	
mm	51	on	/dev/dsk/c5t12d0s0					
mr	52	on	/dev/dsk/c5t12d0s1					
ma	60	on	qfs4	m---2---r-	90%	70%	/qfs4	

EXAMPLE 13-7 samu fDisplay (Continued)

```

mm 61 on /dev/dsk/c5t13d0s0
mr 62 on /dev/dsk/c5t13d0s1
ma 100 on shareqfs1 m---2c--r- 80% 70% /shareqfs1 spade
mm 101 on /dev/dsk/c6t50020F2300004655d0s0
mr 102 on /dev/dsk/c6t50020F2300004655d0s1
ma 110 on shareqfs2 m---2c--r- 80% 70% /shareqfs2 spade
mm 111 on /dev/dsk/c6t50020F2300004655d0s6
mr 112 on /dev/dsk/c6t50020F2300004655d0s7

```

Field Descriptions

The following table shows the field descriptions for this display.

Field	Description
ty	Device type.
eq	Equipment number of the device.
state	<p>Current operating state of the device. Valid device states are as follows:</p> <ul style="list-style-type: none"> ■ on The device is available for access. ■ ro The device is available for read-only access. ■ off The device is not available for access. ■ down The device is available only for maintenance access. ■ idle The device is not available for new operations. Operations in progress continue until completion. ■ nalloc The nalloc flag has been set, which prohibits any future allocation to this device. For more information, see “Per-Logical Unit Number (LUN) Allocation Control” on page 38.
device_name	File system name or path to the device.
status	Device status. For a description of status codes, see “Status Codes” on page 212.
high	High disk usage threshold percentage.
low	Low disk usage threshold percentage.
mountpoint	Mount point of the file system.

Field	Description
server	Name of the host system upon which the file system is mounted.

(F) - Optical Disc Label Display

The F display shows the label on an optical disc.

This display is designed for debugging. It is intended to be used only with the assistance of an Oracle support staff person.

(h) - Help Display

The h display shows a summary of the samu displays available. By default, this is the first display that the system presents when you enter the samu command at the command line.

Navigation

The following table shows the control keys you can use in this display.

Key	Function
Ctrl-B	Page backward
Ctrl-D	Page forward (top portion)
Ctrl-F	Page forward
Ctrl-U	Page backward (top portion)
Ctrl-K	Toggle path display

Sample Display

The online Help has several pages of help screens. On a Sun QFS file system, not all of the displays appear in the initial help screen. For example, the removable media displays are not available if you are running a Sun QFS system. The following example shows only the first page of the SAM-QFS configuration. Subsequent help screens show samu commands.

EXAMPLE 13-8 samu Initial Help Screen for a Sun SAM System

```

Help information   page 1/15   samu 5.0   08:18:13 Sept 8 2008
Displays:
  a  Archiver status           w  Pending stage queue
  c  Device configuration      C  Memory
  d  Daemon trace controls     D  Disk volume dictionary

```

EXAMPLE 13-8 samu Initial Help Screen for a Sun SAM System (Continued)

```

f   File systems
h   Help information
l   Usage information
m   Mass storage status
n   Staging status
o   Optical disk status
p   Removable media load requests
r   Removable media
s   Device status
t   Tape drive status
u   Staging queue
v   Robot catalog

F   Optical disk label
I   Inode
J   Preview shared memory
K   Kernel statistics
L   Shared memory tables
M   Shared memory
N   File system parameters
P   Active Services
R   SAM-Remote
S   Sector data
T   SCSI sense data
U   Device table

more (ctrl-f)

```

(I) - Inode Display

The I display shows the content of inodes.

- To display inodes for an entire file system:

I *filesystem*

For *filesystem*, specify the name of a file system.

- To display a specific inode:

I *inode-number*

For *inode-number*, specify the inode number in either hexadecimal or decimal.

Navigation

The following table shows the control keys you can use in this display.

Key	Function
Ctrl-B	Previous inode
Ctrl-F	Next inode
Ctrl-K	Advance display format

This display is designed for debugging. It is intended to be used only with the assistance of an Oracle support staff person.

Sample Display

The following example shows the inode display.

EXAMPLE 13-9 samuIDisplay

```

Inode      0x1 (1) format: file          samu      5.0 08:27:14 Sept 8 2008
      incore: y

00008100 mode      -r-----          409cdf57 access_time
00000001 ino        (1)                1d32ea20
00000001 gen        (1)                4096b499 modify_time
00000002 parent.ino (2)                02588660
00000002 parent.gen (2)                4096b499 change_time
00000000 size_u          02588660
000c0000 size_l      (786432)          4096b443 creation_time
01000000 rm:media/flags          409a8a7c attribute_time
00000000 rm:file_offset          409c0ce6 residence_time
00000000 rm:mau                00000000 unit/cs/arch/flg
00000000 rm:position          00000000 ar_flags
00000000 ext_attrs  -----          00000000 stripe/stride/sg
00000000 ext.ino   (0)                00000000 media -- --
00000000 ext.gen   (0)                00000000 media -- --
00000000 uid       root                00000000 psize      (0)
00000000 gid       root                000000c0 blocks   (192)
00000001 nlink     (1)                00000600 free_ino  (1536)
00011840 status -n-----  -----

```

Extents (4k displayed as 1k):

```

00_ 000000d0.00 000000e0.00 000000f0.00 00000100.00 00000110.00 00000120.00
06_ 00000130.00 00000140.00 00000150.00 00000160.00 00000170.00 00000180.00
12_ 00000190.00 000001a0.00 000001b0.00 000001c0.00 00000630.00 00000000.00
18_ 00000000.00

```

(J) - Preview Shared Memory Display

The J display shows the shared memory segment for the preview queue.

Navigation

The following table shows the control keys you can use in this display.

Key	Function
Ctrl-B	Page backward
Ctrl-D	Half-page forward
Ctrl-F	Page forward
Ctrl-U	Half-page backward

This display is designed for debugging. It is intended to be used only with the assistance of an Oracle support staff person.

Sample Display

The following example shows the preview shared memory display. This sample output has been truncated.

EXAMPLE 13-10 samu J Display

```
Preview shared memory   size: 155648   samu 5.0 08:30:05 Sept 8 2008

00000000 00040000 00014d58 00000000 00000000 .....MX.....
00000010 00000000 00000000 73616d66 73202d20 .....samfs -
00000020 70726576 69657720 6d656d6f 72792073 preview memory s
00000030 65676d65 6e740000 00026000 00000000 egment.....
00000040 00025fff 00000000 00040000 00014d58 .._.....MX
00000050 00000000 00000000 00000000 00000000 .....
00000060 0000d9e0 00000064 00000000 000001b8 ..Y....d.....8
00000070 3f800000 447a0000 0000d820 00000008 ?...Dz...X ....
```

(K) - Kernel Statistics Display

The K display shows kernel statistics, such as the number of inodes currently in memory.

Navigation

The following table shows the control keys you can use in this display.

Key	Function
Ctrl-B	Page backward
Ctrl-F	Page forward

This display is designed for debugging. It is intended to be used only with the assistance of an Oracle support staff person.

Sample Display

The following example shows the kernel statistics display.

EXAMPLE 13-11 samu K Display

```
Kernel statistics           samu 5.0   08:33:19 Sept 8 2008

module: sam-qfs name: general instance: 0 class: fs
version           4.4.sam-qfs, gumball 2004-05-07 12:12:04
configured file systems      8
mounted file systems        8
nhino                     16384
ninodes                   129526
inocount                  129527
```

EXAMPLE 13-11 samu K Display (Continued)

```
inofree                128577
```

(I) - Usage Display

The `l` display shows the usage information for the file system, including the capacity and space used for each library and file system.

Sample Display

The following example shows a sample usage display.

EXAMPLE 13-12 samu l Display

```
Usage information                samu      5.0 08:36:27 Sept 8 2008
hostid = 80e69e6e OS name: SunOS Architecture: SPARC CPUs: 2 (2 online)

library    40: capacity    389.3G    bytes space    291.1G    bytes, usage    25%
library    51: capacity     9.5G     bytes space     9.5G     bytes, usage     0%
library    55: capacity     0        bytes space     0        bytes, usage     0%
library    56: capacity    10.7G    bytes space    10.7G    bytes, usage     0%
library totals: capacity    409.5G    bytes space    311.3G    bytes, usage    24%

filesystem samfs3: capacity    54.5M    bytes space    13.4M    bytes, usage    75%
filesystem samfs4: capacity   319.5M    bytes space   298.0M    bytes, usage     7%
filesystem samfs7: capacity    96.6M    bytes space    69.6M    bytes, usage    28%
filesystem samfs6: capacity     5.0G    bytes space     4.9G    bytes, usage     3%
filesystem samfs8: capacity     5.0G    bytes space     4.9G    bytes, usage     2%
filesystem totals: capacity   10.5G    bytes space    10.2G    bytes, usage     3%
```

Note – In versions of the software before 4.3, this display showed license information for the file system.

(L) - Shared Memory Tables

The `L` display shows the location of the shared memory tables. It also shows some system defaults that are kept in shared memory.

This display is designed for debugging. It is intended to be used only with the assistance of an Oracle support staff person.

Sample Display

The following shows the shared memory tables.

EXAMPLE 13-13 samu L Display

```

Shared memory tables                                samu 5.0 08:38:31 May 8 2008

shm ptr tbl:
size          12000 (73728)
left          44c8 (17608)
scanner pid   1861
fifo path     01b0 /var/opt/SUNWsamfs/previews
dev_table     01cc
first_dev     0450
scan_mess     cf50
preview_shmid 1
flags         0x20000000
preview stages 55776
preview avail 100
preview count 0
preview sequence 445
age factor    1
fs tbl ptr   0xd820
fseq 10 samfs1 state 0 0 0 0 0
fseq 20 samfs2 state 0 0 0 0 0
fseq 30 qfs1 state 0 0 0 0 0
fseq 40 qfs2 state 0 0 0 0 0
fseq 50 qfs3 state 0 0 0 0 0
fseq 60 qfs4 state 0 0 0 0 0
fseq 100 shareqfs1 state 0 0 0 0 0
fseq 110 shareqfs2 state 0 0 0 0 0

defaults:
optical      mo
tape         lt
timeout      600
stages       1000
log_facility 184
dio minfilesize 100
label barcode FALSE
barcodes low FALSE
export unavail FALSE
attended     TRUE
start rpc    FALSE

vsf factor   1000
fs count     8

```

(m) - Mass Storage Status Display

The `m` display shows the status of mass storage file systems and their member drives. This display shows only mounted file systems.

Sample Display

The following example shows the mass storage status display. Member drives are indented one space and appear directly below the file system to which they belong.

EXAMPLE 13-14 samu m Display

```

Mass storage status                                samu 5.0      08:41:11 Sept 8 2008

ty   eq  status      use state ord  capacity   free   ra  part high low
ms   10  m----2----d  1% on      0    68.354G   68.343G  1M  16  90% 70%
  md   11                1% on      0    13.669G   13.666G
  md   12                1% on      1    13.669G   13.667G
  md   13                1% on      2    13.669G   13.667G
  md   14                1% on      3    13.674G   13.672G
  md   15                1% on      4    13.674G   13.672G
ms   20  m----2----d  1% on      0    68.354G   68.344G  1M  16  90% 70%
  md   21                1% on      0    13.669G   13.667G
  md   22                1% on      1    13.669G   13.667G
  md   23                1% on      2    13.669G   13.667G

```

EXAMPLE 13-14 samu m Display (Continued)

```

md      24          1% on      3  13.674G  13.672G
md      25          1% on      4  13.674G  13.672G
ma     30  m----2----d  4% on      0  64.351G  61.917G      1M   16  90% 70%
mm     31          1% on      0   4.003G   3.988G [8363840 inodes]
md     32          4% on      1  64.351G  61.917G
ma     40  m----2----d  1% on      0  64.351G  64.333G      1M   16  90% 70%
mm     41          1% on      0   4.003G   3.997G [8382784 inodes]
md     42          1% on      1  64.351G  64.333G
ma     50  m----2---r-  1% on      0  64.351G  64.333G      1M   16  90% 70%
mm     51          1% on      0   4.003G   3.997G [8382784 inodes]
mr     52          1% on      1  64.351G  64.333G
ma     60  m----2---r-  1% on      0  64.351G  64.331G      1M   16  90% 70%
mm     61          1% on      0   4.003G   3.997G [8382784 inodes]
mr     62          1% on      1  64.351G  64.331G
ma    100  m----2c--r-  2% on      0 270.672G 265.105G      1M   16  80% 70%
mm    101          1% on      0   2.000G   1.988G [4168992 inodes]
mr    102          2% on      1 270.672G 265.469G
ma    110  m----2c--r-  3% on      0 270.656G 263.382G      1M   16  80% 70%
mm    111          1% on      0   2.000G   1.987G [4167616 inodes]
mr    112          2% on      1 270.656G 264.736G

```

Field Descriptions

The following table shows the field descriptions for this display.

Field	Description
ty	Device type.
eq	Equipment number of the mass storage device.
status	Device status. For a description of status codes, see “Status Codes” on page 212 .
use	Percentage of disk space in use.
state	Current operating state of the mass storage device.
ord	Ordinal number of the disk device within the storage family set.
capacity	Number of 1024-byte blocks of usable space on the disk.
free	Number of 1024-byte blocks of disk space available.
ra	Read-ahead size in kilobytes.
part	Partial stage size in kilobytes.
high	High disk usage threshold percentage.
low	Low disk usage threshold percentage.

(M) - Shared Memory Display

The M display shows the raw shared memory segment in hexadecimal. This is a device table.

Navigation

The following table shows the control keys you can use in this display.

Key	Function
Ctrl-B	Page backward
Ctrl-D	Half-page forward
Ctrl-F	Page forward
Ctrl-U	Half-page backward

This display is designed for debugging. It is intended to be used only with the assistance of an Oracle support staff person.

Sample Display

The following example shows the shared memory display. The sample output has been truncated.

EXAMPLE 13-15 samu M Display

```
Shared memory      size: 73728          samu 5.0      08:43:20 May  8 2008

00000000 00040000 00014d58 00000000 00000000  .....MX.....
00000010 00000000 00000000 73616d66 73202d20  .....samfs -
00000020 73686172 6564206d 656d6f72 79207365  shared memory se
00000030 676d656e 74000000 00012000 000044c8  gment.....DH
00000040 0000dd20 00000000 00000742 00000745  ..] .....B...E
00000050 00000001 00000000 00000000 c0000000  .....@...
00000060 00000001 0001534d 00000000 00000000  .....SM.....
00000070 00000000 00000000 00000000 00000000  .....

00000080 00000000 00000000 00000000 00000000  .....
00000090 20000000 000001b0 000001cc 00000450  .....0...L...P
000000a0 0000cf50 00000001 00000001 4c696365  ..OP.....Lic
000000b0 6e73653a 204c6963 656e7365 206e6576  nse: License nev
000000c0 65722065 78706972 65732e00 00000000  er expires.....
000000d0 00000000 00000000 00000000 00000000  .....
000000e0 00000000 00000000 00000000 00000000  .....
000000f0 00000000 00000000 00000000 00000000  .....
```

(n) - Staging Status Display

The `n` display shows the status of the stager for all media. It displays a list of outstanding stage requests. To display the staging status for a specific media type:

```
n mt
```

For *mt*, specify one of the media types shown in the `mcf(4)` man page.

Sample Display

The following example shows the staging status display.

EXAMPLE 13-16 `samu n` Display

```
Staging status                samu 5.0      08:47:16 May  8 2008

Log output to: /var/opt/SUNWsamfs/stager/log

Stage request: dt.DAT001
Loading VSN DAT001

Staging queues
ty pid  user      status  wait  files  vsn
dt 16097 root      active  0:00   12   DAT001
```

(N) - File System Parameters Display

The `N` display shows all mount point parameters, the superblock version, and other file system information.

Navigation

The following table shows the control keys you can use in this display.

Key	Function
Ctrl-B	Previous file system
Ctrl-D	Page partitions forward
Ctrl-F	Next file system
Ctrl-I	Detailed status interpretations
Ctrl-U	Page partitions backward

This display is designed for debugging. It is intended to be used only with the assistance of an Oracle support staff person.

Sample Display

The following example shows the file system parameters display.

EXAMPLE 13-17 samu N Display

```
File system parameters          samu 5.0      08:55:19 Sept 8 2008

mount_point      : /sam1          partial       : 16k
fs_type          : 6              maxpartial   : 16k
server           :                partial stage : 16384
filesystem name: samfs1          flush_behind : 0
eq_type          : 10 ms          stage_flush_beh: 0
state version    : 0 2            stage_n_window : 262144
(fs,mm)_count    : 5 0            stage_retries  : 3
sync_meta        : 0              stage timeout  : 0
stripe           : 0              dio_consec r,w : 0 0
mm_stripe        : 1              dio_frm_min r,w: 256 256
high low         : 90% 70%        dio_ill_min r,w: 0 0
readahead        : 1048576        ext_bsize     : 4096
writebehind      : 524288
wr_throttle      : 16777216
rd_ino_buf_size : 16384
wr_ino_buf_size  : 512
config           : 0x08520530      mflag         : 0x00000044
status           : 0x00000001

Device configuration:
ty  eq state  device_name          fs family_set
md  11 on    /dev/dsk/c5t8d0s3          10 samfs1
md  12 on    /dev/dsk/c5t8d0s4          10 samfs1
md  13 on    /dev/dsk/c5t8d0s5          10 samfs1
md  14 on    /dev/dsk/c5t8d0s6          10 samfs1
md  15 on    /dev/dsk/c5t8d0s7          10 samfs1
```

(o) - Optical Disc Status Display

The o display shows the status of all optical disk drives configured within the environment.

Navigation

The following table shows the control keys you can use in this display.

Key	Function
Ctrl-B	Page backward
Ctrl-F	Page forward
Ctrl-D	Half-page forward
Ctrl-K	Select (manual, automated library, both, priority)
Ctrl-U	Half-page backward

Sample Display

The following example shows the optical disc drive status display.

EXAMPLE 13-18 samu o Display

```
Optical disk status          samu   5.0   Thu Oct 11 13:15:40
ty eq status      act use state vsn
mo 35 --l---wo-r  1 29% ready oper2
```

Field Descriptions

The following table shows the field descriptions for this display.

Field	Description
ty	Device type.
eq	Equipment number of the optical disc drive.
status	Device status. For a description of status codes, see “Status Codes” on page 212 .
act	Activity count.
use	Percentage of cartridge space used.
state	Current operating state of the optical disc drive. Valid device states are as follows: <ul style="list-style-type: none"> ▪ ready The device is on, and the disc is loaded in the transport; available for access. ▪ notrdy The device is on, but no disc is present in the transport. ▪ idle The device is not available for new connections. Operations in progress continue until completion. ▪ off The device is not available for access. ▪ down The device is available only for maintenance access.
vsn	Volume serial name assigned to the optical disc, or the keyword noLabel if the volume is not labeled.

(p) - Removable Media Load Requests Display

The p display lists information about pending load requests for removable media. You can use the *mt* argument to select either a specific type of media, such as DLT tape, or a family of media, such as tape. The priority display lists the priority in the preview queue, rather than the user queue, and sorts the entries by priority.

The displays shows mount requests in the following formats:

- Both manual and automated library requests by user
- Both manual and automated library requests by priority
- Manual requests only
- Automated library requests only

To display mount requests for devices of a given removable media type:

p *mt*

For *mt*, specify one of the media types shown in the `mc f(4)` man page.

Navigation

The following table shows the control keys you can use in this display.

Key	Function
Key	Function
Ctrl-B	Page backward
Ctrl-D	Half-page forward
Ctrl-F	Page forward
Ctrl-K	Toggle between the different display formats
Ctrl-U	Half-page backward

Sample Display

The following example shows the removable media load requests display.

EXAMPLE 13-19 `samu p` Display

```
Removable media load requests all both samu 5.0 09:14:19 Sept 8 2008
count: 1
```

```
index type pid user rb flags wait count vsn
0 dt 15533 root 150 W--f--- 0:00 DAT001
```

Field Descriptions

The following table shows the field descriptions for this display.

Field	Description
index	Index number in the preview table.
type	Device type code assigned to the removable media.

Field	Description
pid	UNIX process identifier. A process identifier of 1 indicates NFS access.
user	Name assigned to the user requesting the load.
priority	Priority of the request.
rb	Equipment number of the automated library in which the requested VSN resides.
flags	Flags for the device. See the Table 13–1 table.
wait	The elapsed time since the mount request was received.
count	The number of requests for this VSN, if it is a stage.
vsn	Volume serial name of the volume.

Flags

The following table shows the flags for the p display.

TABLE 13–1 Flag Values for the samu p Display

Flag	Description
W-----	Write access requested
}}>{{}b----	Entry is busy
-C---	Clear VSN requested
--f--	File system requested
---N-	Media is foreign to the file system
---S{-}	Flip side already mounted
-----s	Stage request flag

(P) - Active Services Display

The P display lists the services registered with the Sun QFS single port multiplexer.

Navigation

The following table shows the control keys you can use in this display.

Key	Function
Ctrl-B	Page backward
Ctrl-F	Page forward

This display is designed for debugging. It is intended to be used only with the assistance of an Oracle support staff person.

Sample Display

The following example shows the active services display.

EXAMPLE 13-20 samu P Display

```
Active Services                samu      5.0      09:08:33 Sept 8 2008

Registered services for host &rdquo;pup&rsquo;;:
  sharedfs.qfs2
  sharedfs.qfs1
  2 services registered.
```

(r) - Removable Media Status Display

The r display monitors the activity on removable media devices such as tape drives. You can monitor either a specific type of device, such as video tape, or a family of devices such as all tape devices.

To display the status for a specific device:

```
r eq
```

For *eq*, specify the equipment number for the device.

Sample Display

The following example shows the removable media status display.

EXAMPLE 13-21 samu r Display

```
Removable media status: all          samu 5.0      09:11:27 Sept 8 2008

ty  eq  status      act  use  state  vsn
dt 150  --l-----r   0  63%  ready  DAT001
```

Field Descriptions

The following table shows the field descriptions for this display.

Field	Description
ty	Device type.
eq	Equipment number of the drive.
status	Device status. For a description of status codes, see “Status Codes” on page 212 .
act	Activity count.
use	Percentage of cartridge space used.
state	<p>Current operating state of the removable media. Valid device states are as follows:</p> <ul style="list-style-type: none"> ▪ <code>ready</code> The device is on, and the disk or tape is loaded in the transport available for access. ▪ <code>not rdy</code> The device is on, but no disk or tape is present in the transport. ▪ <code>idle</code> The device is not available for new connections. Operations in progress continue until completion. ▪ <code>off</code> The device is not available for access. ▪ <code>down</code> The device is available only for maintenance access. ▪ <code>naLloc</code> The <code>naLloc</code> flag has been set, which prohibits any future allocation to this device. For more information, see “Per-Logical Unit Number (LUN) Allocation Control” on page 38.
vsn	Volume serial name assigned to the volume, or the keyword <code>noLabel</code> if the volume is not labeled. Blank if no volume is present in the transport, or device is off.

(R) - Sun SAM-Remote Information Display

The R display shows information and status on Sun SAM-Remote configurations.

This display is designed for debugging. It is intended to be used only with the assistance of an Oracle support staff person.

(s) - Device Status Display

The s display shows the status for all devices configured within the environment.

Navigation

The following table shows the control keys you can use in this display.

Key	Function
Ctrl-B	Page backward
Ctrl-D	Half-page forward
Ctrl-F	Page forward
Ctrl-U	Half-page backward

Sample Display

The following example shows the device status display.

EXAMPLE 13-22 samus Display

```
Device status                    samu      5.0
09:14:05 Sept 8 2008

ty   eq   state  device_name                fs  status      pos
sk   100  on      /etc/opt/SUNWsamfs/dcstkconf 100 m-----r
stk_dismount / (2275) 0, volser 700073
sg   120  on      /dev/rmt/2cbn                100 -----p  empty
sg   130  on      /dev/rmt/5cbn                100 --l----o-r
Ready for data transfer
sg   140  on      /dev/rmt/6cbn                100 -----p  empty
sg   150  on      /dev/rmt/4cbn                100 -----p  empty
hy   151  on      historian                     151 -----
```

Field Descriptions

The following table shows the field descriptions for this display.

Field	Description
ty	Device type.
eq	Equipment number of the device.
state	Current operating state of the device.
device_name	Path to the device. For file system devices, this is the file system name.
fs	Equipment number of the family set to which the device belongs.
status	Device status. For a description of status codes, see “Status Codes” on page 212 .

(S) - Sector Data Display

The S display shows raw device data.

Navigation

The following table shows the control keys you can use in this display.

Key	Function
Ctrl-B	Previous sector
Ctrl-D	Page forward (top portion)
Ctrl-F	Next sector
Ctrl-K	Advance display format
Ctrl-U	Page backward (top portion)

This display is designed for debugging. It is intended to be used only with the assistance of an Oracle support staff person.

(t) - Tape Drive Status Display

The t display shows the status of all tape drives configured within the environment.

Navigation

The following table shows the control keys you can use in this display.

Key	Function
Ctrl-B	Page backward
Ctrl-F	Page forward

Sample Display

The following example shows the tape drive status display.

EXAMPLE 13-23 samu t Display

```
Tape drive status                samu    5.0    09:21:07 Sept 8 2008

ty  eq  status      act  use  state  vsn
sg 120  -----p    0   0%  notrdy
sg 130  -----p    0   0%  notrdy
sg 140  -----p    0   0%  notrdy
sg 150  --l-----r    0  41%  ready  700088 idle
```

Field Descriptions

The following table shows the field descriptions for this display.

Field	Description
ty	Device type.
eq	Equipment number of the drive.
status	Device status. For a description of status codes, see “Status Codes” on page 212 .
act	Activity count.
use	Percentage of cartridge space used.
state	<p>Current operating state of the removable media. Valid device states are as follows:</p> <ul style="list-style-type: none"> ▪ <code>ready</code> The device is on and the disk or tape is loaded in the transport; available for access. ▪ <code>not rdy</code> The device is on but no disk or tape is present in the transport. ▪ <code>idle</code> The device is not available for new connections. Operations in progress continue until completion. ▪ <code>off</code> The device is not available for access. ▪ <code>down</code> The device is available only for maintenance access. ▪ <code>nal loc</code> The <code>nal loc</code> flag has been set, which prohibits any future allocation to this device. For more information, see “Per-Logical Unit Number (LUN) Allocation Control” on page 38.
vsn	Volume serial name assigned to the volume, or the keyword <code>no label</code> if volume is not labeled. Blank if no volume is present in the transport, or device is off.

(T) - SCSI Sense Data Display

The T display shows the SCSI status of a SCSI device.

Navigation

The following table shows the control keys you can use in this display.

Key	Function
Ctrl-B	Previous equipment
Ctrl-F	Next equipment

This display is designed for debugging. It is intended to be used only with the assistance of an Oracle support staff person.

(u) - Staging Queue Display

The u display lists all files in the staging queue.

Navigation

The following table shows the control keys you can use in this display.

Key	Function
Ctrl-B	Page backward
Ctrl-D	Half-page forward
Ctrl-F	Page forward
Ctrl-K	Display the path on the second line of each entry
Ctrl-U	Half-page backward

Sample Display

The following example shows the staging queue display.

EXAMPLE 13-24 samu u Display

```
Staging queue by media type: all          samu 5.0      09:24:23 Sept 8 2008
      volumes 1 files 22
```

ty	length	fseq	ino	position	offset	vsn
dt	451.611k	20	1030	207cc	473	DAT001
dt	341.676k	20	1031	207cc	7fc	DAT001
dt	419.861k	20	1032	207cc	aa9	DAT001
dt	384.760k	20	1033	207cc	df2	DAT001
dt	263.475k	20	1034	207cc	10f5	DAT001
dt	452.901k	20	1035	207cc	1305	DAT001
dt	404.598k	20	1036	207cc	1690	DAT001
dt	292.454k	20	1037	207cc	19bb	DAT001
dt	257.835k	20	1038	207cc	1c05	DAT001
dt	399.882k	20	1040	207cc	1e0b	DAT001
dt	399.882k	40	1029	208d7	2	DAT001
dt	257.835k	40	1030	208d7	323	DAT001
dt	292.454k	40	1031	208d7	528	DAT001
dt	404.598k	40	1032	208d7	772	DAT001
dt	452.901k	40	1033	208d7	a9d	DAT001
dt	263.475k	40	1034	208d7	e28	DAT001
dt	384.760k	40	1035	208d7	1038	DAT001
dt	419.861k	40	1036	208d7	133b	DAT001

EXAMPLE 13-24 samu u Display (Continued)

```
dt 341.676k 40 1037 208d7 1684 DAT001
dt 451.611k 40 1038 208d7 1931 DAT001
dt 161.326k 40 1039 208d7 1cba DAT001
dt 406.400k 40 1040 208d7 1dfe DAT001
```

Field Descriptions

The following table shows the field descriptions for this display.

Field	Description
ty	Device type
length	File length
fseq	File system equipment number
ino	The inode number
position	The position of the archive file on the specific medium
offset	Offset of the archive file on the specific medium
vsn	Volume serial name of the volume

(U) - Device Table Display

The U display shows the device table in a human-readable form.

To display the device table for a specific device:

U *eq*

For *eq*, specify the equipment number of the device.

Navigation

The following table shows the control keys you can use in this display.

Key	Function
Ctrl-B	Previous equipment
Ctrl-F	Next equipment

This display is designed for debugging. It is intended to be used only with the assistance of an Oracle support staff person.

Sample Display

The following example shows the device table display.

EXAMPLE 13-25 samu U Display

```
Device table: eq: 10      addr: 00000450  samu 5.0      09:28:40 Sept 8 2008
```

message:

```
0004000000014d58 0000000000000000      00000000 delay
0000000000000000 mutex                  00000000 unload_delay
00000aa8 next
73616d66 set:  samfs1
73310000
00000000
00000000
000a000a eq/fseq
08010801 type/equ_type
0000      state
00000000 st_rdev
00000000 ord/model
00000000 mode_sense
00000000 sense
00000000 space
00000000 capacity
00000000 active
00000000 open
00000000 sector_size
00000000 label_address
00000000 vsn:
00000000
00000000
00000000
00000000 status: -----
00000000 dt
73616d66 name: samfs1
```

(v) - Automated Library Catalog Display

The v display shows the location and VSN of all disks or tapes currently catalogued in the automated library.

To display catalog information for a specific device:

```
v eq
```

For *eq*, specify the equipment number of the device. Type the keyword *historian* to view the historian catalog.

Navigation

The following table shows the control keys you can use in this display.

Key	Function
Ctrl -B	Page backward.
Ctrl -D	Next library catalog.
Ctrl -F	Page forward.
Ctrl -I	Detailed, two-line display format. When you enter Ctrl -I once, it shows times and barcodes. When you enter Ctrl -I a second time, it shows volume reservations on the second line.
Ctrl -K	Advance sort key. After you enter Ctrl-K, you can type one of the following numbers to select a sort key: <ul style="list-style-type: none"> 1 - Sort by slot 2 - Sort by count 3 - Sort by usage 4 - Sort by VSN 5 - Sort by access time 6 - Sort by barcode 7 - Sort by label time
Ctrl-U	Previous automated library catalog.
/	Search for VSN.
%	Search for barcode.
\$	Search for slot.

Sample Display

The following example shows the automated library catalog display.

EXAMPLE 13-26 samu v Display

```
Robot VSN catalog by slot      : eq 100samu    5.0    09:30:25 Sept 8 2008
count 32
slot      access time count use flags      ty vsn
0      2004/05/08 08:35 64 0% -il-o-b----- sg 700071
1      2004/05/08 09:08 27 12% -il-o-b----- sg 700073
2      2004/05/08 09:12 26 12% -il-o-b----- sg 700077
3      2004/05/08 08:39 37 40% -il-o-b----- sg 700079
4      2004/05/08 09:16 24 6% -il-o-b----- sg 700084
5      2004/05/08 09:18 24 41% -il-o-b----- sg 700088
6      none                0 0% -il-o-b----- sg 700090
7      none                0 0% -il-o-b----- sg 700092
```

EXAMPLE 13-26 samu v Display (Continued)

```

      8      none      0  0% -il-o-b----- sg 000155
      9      none      0  0% -il-o-b----- sg 000156
     10      none      0  0% -il-o-b----- sg 000157
     11      none      0  0% -il-o-b----- sg 000158
     12      none      0  0% -il-o-b----- sg 000154
     13      none      0  0% -il-o-b----- sg 000153
     14      none      0  0% -il-o-b----- sg 000152

```

Field Descriptions

The following table shows the field descriptions for this display.

TABLE 13-2 samu v Display Field Descriptions

Field	Description
Robot VSN catalog	Name of the specified automated library and time the display refreshed.
count	Number of slots allocated in this library's catalog.
slot	Slot number within the specified library.
access time	Time the volume was last accessed.
count	Number of accesses to this volume since the last audit operation.
use	Percentage of space used for the volume.
flags	Flags for the device. See the table in the Table 13-3 section for information about the flags.
ty	Device type.
vsn	Volume serial name of the volume.
LVTTime	Time stamp of when the entire tape was last verified.
LVPos	Position where the <code>tpverify</code> command was canceled.

Flags

The following table shows the flags using in the `flags` field. In some cases, more than one flag can occur in a field, and one flag overrides the other.

TABLE 13-3 Flags Values for samu v Display

Flags	Description
A-----	Volume needs audit.
}}{{{i}-----	Slot in use.

TABLE 13-3 Flags Values for samu v Display (Continued)

Flags	Description
-l-----	Labeled. Overrides N.
-N-----	Unlabeled. This volume is foreign to the environment.
--E-----	Media error. Set when the software detects a write error on a cartridge.
--o-----	Slot occupied.
---C----	Volume is a cleaning tape. Overrides p.
---p----	Priority VSN.
---b----	Barcode detected.
----W--	Write protect. Set when the physical write protection mechanism is enabled on a cartridge.
-----R{-}-	Read only.
-----c{-}-	Recycle.
-----d{-}	Duplicate VSN. Overrides U.
-----U{-}	Volume unavailable.
-----f	Archiver found volume full.
-----X	Export slot.

(w) - Pending Stage Queue

The w display shows queued stage requests for which the volumes have not yet been loaded.

To display the pending stage queue for a specific media type:

```
w mt
```

For *mt*, specify one of the media types shown in the `mc f(4)` man page.

Navigation

The following table shows the control keys you can use in this display.

Key	Function
Ctrl-B	Page backward
Ctrl-D	Half-page forward

Key	Function
Ctrl-F	Page forward
Ctrl-K	Display the path on the second line of each entry
Ctrl-U	Half-page backward

Sample Display

The following example shows the pending stage queue display.

EXAMPLE 13-27 samu w Display

```
Pending stage queue by media type: all      samu      5.0      Thu Oct 11 13:20:27
      volumes 1 files 13
```

ty	length	fseq	ino	position	offset	vsn
at	1.383M	1	42	3a786	271b	000002
at	1.479M	1	56	3a786	5139	000002
at	1018.406k	1	60	3a786	6550	000002
at	1.000M	1	65	3a786	7475	000002
at	1.528M	1	80	3a786	99be	000002
at	1.763M	1	92	3a786	ce57	000002
at	1.749M	1	123	3a786	11ece	000002
at	556.559k	1	157	3a786	1532f	000002
at	658.970k	1	186	3a786	17705	000002
at	863.380k	1	251	3a786	1dd58	000002
at	1.268M	1	281	3a786	1f2b7	000002
at	1.797M	1	324	3a786	23dfa	000002
at	1.144M	1	401	3a786	2bb6d	000002

Field Descriptions

The following table shows the field descriptions for this display.

Field	Description
ty	Device type
length	File length
fseq	File system equipment number
ino	The inode number
position	The position (in decimal format) of the archive file on the specific medium
offset	Offset of the archive file on the specific medium
vsn	Volume serial name of the volume

Status Codes

The operator displays have different status codes for the removable media device displays and the file system displays. The following sections describe these status codes.

Removable Media Device Display Status Codes

The o, r, s, and t operator displays show status codes for removable media devices. Status codes are displayed in a 10-position format, reading from left (position 1) to right (position 10).

These status codes do not apply to the samu f, m, and v displays. For information about the status codes for the f and m displays, see [Table 13–5](#). For information about the status codes for the v display, see “(v) - Automated Library Catalog Display” on page 207.

The following table defines the status codes for each position.

TABLE 13–4 Status Code for Removable Media Device Display

Status Bit	Meaning for a Device
s-----	Media is being scanned
m-----	The automated library is operational
M-----	Maintenance mode
}}{{{E}-----	Device received an unrecoverable error in scanning
}}{{{a}-----	Device is in audit mode
-l-----	Media has a label
-N-----	Foreign media
-L-----	Media is being labeled
--I-----	Waiting for device to idle
--A-----	Needs operator attention
--C---	Needs cleaning
--U---	Unload has been requested
---R--	Device is reserved
----w{-}-	A process is writing on the media
----o{-}-	Device is open
-----P{-}	Device is positioning (tape only)
-----F{-}	For automated libraries, all storage slots occupied. For tape and magneto-optical drives, media is full

TABLE 13-4 Status Code for Removable Media Device Display (Continued)

Status Bit	Meaning for a Device
-----R	Device is ready and the media is read-only
-----r	Device is spun up and ready
-----p	Device is present
-----W	Device is write protected

File System Display Status Codes

The *f* and *m* operator displays show status codes for file systems. Status codes are displayed in an 11-position format, reading from left (position 1) to right (position 11).

These status codes do not apply to the *samu c*, *o*, *r*, *s*, *t*, or *v* displays. For information about the status codes for the *o*, *r*, *s*, and *t* displays, see [Table 13-4](#). For information about the status codes for the *v* display, see “(v) - Automated Library Catalog Display” on page 207.

The following table defines the status codes for each position.

TABLE 13-5 File System Display Status Codes

Status Bit	Meaning for a File System
m-----	File system is currently mounted
M-----	File system is being mounted
}}>{{}u-----	File system is being unmounted
-A-----	File system data is being archived
--R-----	File system data is being released
--S----	File system data is being staged
---1---	Sun SAM file system version 1
---2---	Sun SAM file system version 2
----c--	Sun QFS shared file system
----W{-}-	Single writer
-----R{-}-	Multireader
-----r{-}	mr devices
-----d	md devices

States of an Operator Display Device

The `c`, `m`, `o`, `r`, `s`, and `t` operator displays show state codes for each device. These codes represent the current access state for the device.

The following table defines the state codes.

TABLE 13-6 Operator Display Device States

Device State	Description
<code>on</code>	The device is available for access. For certain displays, this state might be superseded by the states <code>ready</code> or <code>not rdy</code> .
<code>ro</code>	The device is available for read-only access. For certain displays, this state might be superseded by the states <code>ready</code> or <code>not rdy</code> .
<code>off</code>	The device is not available for access. For tape and optical disk drives, possible reasons for the device to be in the <code>off</code> state include the following: <ul style="list-style-type: none"> ■ Cleaning was requested but no cleaning cartridge was found in the automated library. ■ The cleaning cartridge cannot be loaded or unloaded from the drive. ■ Initialization found the drive status to be full and attempts to clear the drive failed. ■ The system was unable to clear a cartridge from a drive. ■ Opening the drive for I/O failed during spin-up. ■ An error other than NOT READY was received when spinning down the drive for unloading. ■ Opening the standard tape driver on the drive failed during spin-up.
<code>down</code>	The device is available for maintenance access only.
<code>idle</code>	The device is not available for new connections. Operations in progress continue until completion.
<code>ready</code>	The device is on and the disk or tape loaded in the transport is available for access.
<code>not rdy</code>	The device is on but no disk or tape is present in the transport.
<code>unavail</code>	The device is unavailable for access and cannot be used for automatic operations. You can continue to use the <code>load</code> and the <code>unload</code> commands for moving media while the device is in the <code>unavail</code> state.
<code>nalloc</code>	The <code>nalloc</code> flag has been set, which prohibits any future allocation to this device. For more information, see “Per-Logical Unit Number (LUN) Allocation Control” on page 38.

You can use the `samu down`, `off`, and `on` device state commands to change device states. You can issue these commands from any `samu` display, but if you issue them from the `c`, `m`, `o`, `r`, `s`, or `t` display, you can see the change in the display.

The following procedure show how to change a device's state from down to on.

▼ How to Change a Drive State From down to on

- 1 **Type one of the following character keys to invoke a `samu` display that show drive and automated library device states: `c`, `m`, `o`, `r`, `s`, or `t`.**
- 2 **Choose a device and verify that it is in the down state.**
- 3 **Issue the `off` command to stop all activity for the device.**

command: `off eq`

where `eq` is the equipment number of the device.

- 4 **Issue the `on` command.**

For example:

Command: `on eq`

For `eq`, specify the equipment number of the device.

Operator Commands

The following topics describe the operator commands that you can issue from the `samu` operator utility's command interface. You can issue the commands from any display.

The operator commands types are described in the following sections:

- Device Commands
- File System Commands: I/O Management
- File System Commands: Direct I/O Management
- File System Commands: Sun QFS Shared File Systems
- File System Commands: Miscellaneous
- Miscellaneous Commands

If you want to issue any operator commands from the Oracle Solaris OS command line, you must use them as arguments to the `samcmd` command. For more information about the `samcmd` command, see “[samcmd\(1M\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

Device Commands

The following table shows the device commands and their actions.

TABLE 13-7 Device Command Actions

Command	Action
<code>down</code>	Terminates operation on device <i>eq</i> .
<code>idle</code>	Restricts access to device <i>eq</i> by preventing new connections to the device. Existing operations continue until completion.
<code>off</code>	Logically turns off device <i>eq</i> .
<code>on</code>	Logically turns on device <i>eq</i> .
<code>unavail</code>	Selects device <i>eq</i> and makes it unavailable for use with the file system. You might set a drive state to <code>unavail</code> , for example, in a disaster recovery situation in which you are trying to load media to restore a file system and you do not want the Sun SAM software to attempt to use this drive.
<code>unload</code>	Unloads the mounted media for the specified removable media device <i>eq</i> . For magazine devices, the <code>unload</code> command unloads the mounted cartridge and ejects the magazine.
<code>nalloc</code>	Sets the <code>nalloc</code> flag on the device, which prohibits any allocation to this device. For more information, see “Per-Logical Unit Number (LUN) Allocation Control” on page 38.
<code>alloc</code>	Removes the <code>nalloc</code> flag from the device. The <code>nalloc</code> flag prohibits any allocation to this device. The <code>on</code> command also removes this flag. For more information, see “Per-Logical Unit Number (LUN) Allocation Control” on page 38.

All these commands use the following syntax:

```
:command eq
```

For *eq*, specify the equipment Nnumber of the device.

File System Commands: I/O Management

The commands described in this section enable you to manage I/O characteristics dynamically.

The `flush_behind` Command

The `flush_behind` command sets the maximum `flush_behind` value. When set to a value greater than 0, modified pages that are being written sequentially are written to disk

asynchronously to help the Oracle Solaris kernel layer keep the pages clean. By default, the maximum value is 0, which disables `flush_behind`.

:flush_behind *eq value*

For *value*, specify an integer number of kilobytes such that $0 \leq \text{value} \leq 8192$.

For *eq*, specify the equipment number for the file system.

force_nfs_async and noforce_nfs_async Commands

These commands enable you to control whether the file system caches NFS data written to the server even if NFS has requested that the data be written synchronously through to disk. The `force_nfs_async` command caches NFS data.

The `force_nfs_async` command is effective only if the file system is mounted as an NFS server and only if the clients are mounted with the `noac` NFS mount option. For more information about mounting an NFS file system, see the `mount_nfs(1M)` man page.



Caution – The `force_nfs_async` option violates NFS protocols. Use this command with caution. In the event of a server interruption, data can be lost. Data is cached on the NFS server and cannot be seen immediately by all the clients if there are multiple NFS servers. Multiple NFS servers can be enabled within the Sun QFS shared file system. For more information about the Sun QFS shared file system, see [Chapter 5, “Configuring a Shared File System.”](#)

The `noforce_nfs_async` command, which is the default, synchronously writes data through to disk.

:force_nfs_async *eq*
:noforce_nfs_async *eq*

For *eq*, specify the equipment number for the file system.

readahead Command

The `readahead` command specifies the maximum number of bytes that can be read ahead by the file system. The default *contig* number is 8 (131072 bytes).

:readahead *eq contig*

For *eq*, specify the equipment number for the file system.

For *contig*, specify units of 1-kilobyte blocks. This value must be an integer such that $1 < \text{contig} < 8192$. The *contig* specified is truncated to a multiple of 8 kilobytes.

For example, the following command sets the maximum contiguous block size to 262,144 bytes for the file system defined as equipment number 3:

```
:readahead 3 256
```

This value can also be configured in the `samfs.cmd` file by specifying the `readahead` directive. For more information, see “[samfs.cmd\(4\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

sw_raid and nosw_raid Commands

These commands specify whether the file system aligns the `writebehind` buffer. Specify `sw_raid` if the software RAID feature of a package such as Solaris Volume Manager is also used on this file system. The default setting is `nosw_raid`.

```
:sw_raid eq  
:nosw_raid eq
```

For `eq`, specify the equipment number for a file system.

writebehind Command

The `writebehind` command specifies the maximum number of bytes that can be written behind by a file system. The default `contig` number is 8 (131072 bytes).

```
:writebehind eq contig
```

For `eq`, specify the equipment number for a file system.

For `contig`, specify units of 1-kilobyte blocks. This value must be an integer such that $1 < contig < 8192$.

For example, the following command sets the maximum contiguous block size to 262,144 bytes for the file system defined as equipment number 50:

```
:writebehind 50 256
```

This value can also be configured in the `samfs.cmd` file by specifying the `writebehind` directive. For more information, see “[samfs.cmd\(4\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

wr_throttle Command

The `wr_throttle` command sets the number of outstanding write kilobytes for one file. The default is the number of kilobytes that is 2% of the memory size.

If the percentage cannot be calculated, set the value to 100 MBytes. To determine memory size on Oracle Solaris:

```
npages = sysconf(_SC_PHYS_PAGES);  
pagesizeb = sysconf(_SC_PAGESIZE);
```

```
fprintf(fp, "Memory size:\t\t%.1f MBytes
(%.1f M2Bytes, %ld pages, %ld bytes/page)\n", ( /
(float)npages * (float)pagesize) / 1000000.0, ((float)npages * (float)pagesize) /
/ 1048576.0, npages, pagesizeb);
```

After you have found the memory size, set the value of `wr_throttle` to 2% of the total size, as shown in the following table.

Memory Size	Value for <code>wr_throttle</code>
1 GByte	20 MBytes
4 GBytes	80 MBytes
16 GBytes	320 MBytes
64 GBytes	1.3 GBytes

`:wr_throttle eq value`

For *eq*, specify the equipment number for a file system.

For *value*, specify an integer number of kilobytes. If *value*=0, there is no limit.

File System Commands: Direct I/O Management

The commands described in this section control I/O on Sun QFS file systems. You can change the type of I/O for an individual file based on I/O size and history. If direct I/O is specified for a file, for example, through the `set fa` command, these options are ignored and all I/O to regular files is direct, if possible.

These commands refer to both well-aligned and misaligned I/O.

- Well-aligned I/O occurs when the file offset falls on a 512-byte boundary and when the length of the I/O transfer is at least 512 bytes.
- Misaligned I/O occurs when the file offset does not fall on a 512-byte boundary and the length of the transfer is less than 512 bytes.

For more information about I/O and I/O management, see [Chapter 7, “Advanced File System Topics.”](#)

`dio_rd_form_min` and `dio_wr_form_min` Commands

These commands set the lower limits for well-aligned I/O to the specified number of 1024-byte blocks. Use the `dio_rd_form_min` command to set the number for read operations, and use the `dio_wr_form_min` command to set the number for write operations. By default, the minimum number of blocks is 256.

```
:dio_rd_form_min eq value  
:dio_wr_form_min eq value
```

For *eq*, specify the equipment number for the file system.

For *value*, specify an integer number of 1024-byte blocks to use for the lower limit. If *value=0*, automatic I/O switching is disabled.

dio_rd_ill_min and dio_wr_ill_min Commands

These commands set the lower limit for misaligned I/O to the specified number of 1024-byte blocks. Use the `dio_rd_ill_min` command to set the number for read operations, and use the `dio_wr_ill_min` command to set the number for write operations. By default, the minimum number of blocks is 256.

```
:dio_rd_ill_min eq value  
:dio_wr_ill_min eq value
```

For *eq*, specify the equipment number for the file system.

For *value*, specify an integer number of 1024-byte blocks to use for the lower limit. If *value=0*, automatic I/O switching is disabled.

dio_rd_consec and dio_wr_consec Commands

These commands set the number of consecutive I/O transfers that can occur with a buffer size greater than the specified lower limits. By default, *value=0*, which means that no default direct read operations occur based on I/O sizes.

```
:dio_rd_consec eq value  
:dio_wr_consec eq value
```

For *eq*, specify the equipment number for the file system.

For *value*, specify the number of consecutive I/O transfers with a buffer size greater than the specified lower limit. The specified lower limit is the *value* of `dio_rd_form_min` for aligned read operations or `dio_rd_ill_min` for misaligned read operations.

For more information, see the following sections:

- “`dio_rd_form_min` and `dio_wr_form_min` Commands” on page 219
- “`dio_rd_ill_min` and `dio_wr_ill_min` Commands” on page 220

dio_szero and nodio_szero Commands

These commands set or clear the direct I/O sparse zeroing mount option.

The `dio_szero` command causes uninitialized areas of sparse files written with direct I/O to be zeroed when the area is accessed. This behavior makes the sparse file behavior the same as that for paged I/O. By default, sparse files written by direct I/O do not have the uninitialized areas zeroed for performance reasons. The default is `nodio_szero`.

```
:dio_szero eq
:nodio_szero eq
```

For *eq*, specify the equipment number for the file system.

forcedirectio and noforcedirectio Commands

These commands enable you to control whether direct I/O is used as the default I/O mode. By default, the I/O mode is buffered and uses the page cache. The `forcedirectio` command enables direct I/O for all transfers. The `noforcedirectio` command restores the default, buffered I/O.

When direct I/O is specified, the system transfers data directly between the users buffer and disk. Use direct I/O only if the file system is used for large, block-aligned, sequential I/O.

```
:forcedirectio eq
:noforcedirectio eq
```

For *eq*, specify the equipment number for the file system.

For more information about I/O, see [Chapter 7, “Advanced File System Topics.”](#)

File System Commands: Sun QFS Shared File Systems

The file system commands described in this section are supported on Sun QFS shared file systems only.

meta_timeo Command

The `meta_timeo` command sets the time limit for the Sun QFS shared file system metadata cache. The default value is 3. For more information about using this feature, see [“Retaining Cached Attributes: \(meta_timeo Option\)”](#) on page 169.

```
:meta_timeo eq interval
```

For *eq*, specify the equipment number of the file system.

For *interval*, specify time in seconds. After this interval expires, the client host systems obtain a new copy of the metadata information from the metadata server host.

mh_write and nomh_write Commands

These commands enable or disable multihost read and write operations. For information about this feature, see [“Enabling Multiple Host Reads and Writes: \(mh_write Option\)”](#) on page 167.

```
:mh_write eq  
:nomh_write eq
```

For *eq*, specify the equipment number of the file system.

minallocsz and maxallocsz Commands

These commands set the minimum and maximum block allocation size.

```
:minallocsz eq value  
:maxallocsz eq value
```

For *eq*, specify the equipment number of the file system.

For *value*, and for more information about this feature, see [“Tuning Allocation Sizes: \(minallocsz and maxallocsz Options\)”](#) on page 166.

rdlease, wrlease, and aplease Commands

These commands control the amount of time granted for read, write, and append leases. The default time is 30 seconds. For information about this feature, see [“Using Leases in a Sun QFS Shared File System: \(rdlease, wrlease, and aplease Options\)”](#) on page 166.

```
:rdlease eq interval  
:wrlease eq interval  
:aplease eq interval
```

For *eq*, specify the equipment number of the file system.

For *interval*, specify an integer number of seconds, $15 \leq \text{interval} \leq 600$.

File System Commands: Miscellaneous

The commands described in this section enable you to control leases, allocation sizes, and various other file system characteristics.

abr and noabr Commands

These commands set or clear the application binary recovery (ABR) mount option.

Use these commands only in an Oracle RAC environment with Sun QFS asynchronous I/O (AIO). These mount options disable or enable ABR of software mirrors. They apply only to Sun QFS file systems built on Solaris Volume Manager mirrored volumes that support ABR.

```
:abr eq  
:noabr eq
```

For *eq*, specify the equipment number for the file system.

dmr and nodmr Commands

These commands set or clear the direct mirror reads (DMR) mount option.

Use these commands only in an Oracle Real Application Cluster (RAC) environment with Sun QFS AIO. These mount options disable or enable DMR of software mirrors. They apply only to Sun QFS file systems built on Solaris Volume Manager mirrored volumes that support DMR.

```
:dmr eq
:nodmr eq
```

For *eq*, specify the equipment number for the file system.

invalid interval Command

The `invalid` command specifies that the file system hold cached attributes for at least the specified number of seconds after a file is modified. You can specify this command only if the file system was mounted originally with the `reader` mount option. For information about mount options, see “`mount_samfs(1M)`” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

```
:invalid eq interval
```

For *eq*, specify the equipment number for the file system.

For *interval*, specify the number of seconds to hold the attributes after file modification. For example, assume that `interval=30`. When you issue an `ls` command, you might not see a newly created file appear in its output for 30 seconds after it has been created on its writer host.

mm_stripe Command

The `mm_stripe` command sets the metadata stripe width for the file system to the specified number of 16-kilobyte disk allocation units (DAUs). The default is 1 DAU so that the file system writes one DAU of metadata to one LUN before switching to another LUN.

```
:mm_stripe eq value
```

For *eq*, specify the equipment number of the file system.

For *value*, specify either 0 or 1. If *value* is set to 1, which is the default, the file system writes one DAU of metadata to one LUN before switching to another LUN. If *value* is set to 0, the metadata is round-robin across all available metadata LUNs.

qwrite and noqwrite Commands

The `qwrite` and `noqwrite` commands control the ability to perform simultaneous read and write operations to the same file from different threads. Specify `qwrite` only if file system users handle multiple simultaneous transactions to the same file. This feature is useful in database applications. The `qwrite` feature improves I/O performance by queuing multiple requests at the drive level. The `qwrite` specification is disabled for NFS reads or writes of the file system.

The default setting is `noqwrite`, so the file system disables simultaneous read and write operations to the same file. This mode is defined by the UNIX vnode interface standard that gives exclusive access to only one writer and forces other writers and readers to wait.

```
:qwrite eq
:noqwrite eq
```

For *eq*, specify the equipment number of the file system.

refresh_at_eof and norefresh_at_eof Commands

The `refresh_at_eof` and `norefresh_at_eof` commands can be used for fast updates to Sun QFS hosts that are mounted with the `reader` mount option in a multireader file system. This option ensures that the system refreshes the current file size when the read buffer exceeds the end of file. You can use this option, for example, if the writer host system is appending to a file and the reader is issuing `tail` commands with the `-f` option. The default is `norefresh_at_eof`.

```
:refresh_at_eof eq
:norefresh_at_eof eq
```

For *eq*, specify the equipment number of the file system.

suid and nosuid Commands

The `suid` and `nosuid` commands control whether running programs are allowed to automatically change their owner IDs. For more information about the implications of using these mount options, see the `suid` and `nosuid` mount option descriptions on the `mount_ufs(1M)` man page and the `suid(2)` man page.

```
:suid eq
:nosuid eq
```

For *eq*, specify the equipment number of the file system.

stripe Command

The `stripe` command sets the stripe width for the file system to the specified number of disk allocation units (DAUs). The stripe width specifies that *value* multiplied by the DAU bytes are written to one LUN before switching to the next LUN. You can use the `sammkfs -a` command to set the DAU size on the file system when it is initialized.

```
:stripe eq value
```

For *eq*, specify the equipment number of the file system.

For *value*, specify an integer such that $0 < value < 255$. If *value*=0, files are round-robin on each slice. The default *value* on file systems with an `ms` equipment type and on file systems with an `ma` equipment type with no striped group (`gXXX`) components is as follows:

- 128 kilobytes/DAU for DAUs < 128 kilobytes
- 1 for DAUs > 128 kilobytes

By default, `value=0` on a Sun QFS shared file system.

By default, `value=0` on file systems with an `ma` equipment type with any striped group (`gXXX`) components.

The system sets `value=0` if mismatched striped groups exist.

For more information about file system types, see “Sun QFS File Systems Design Basics” on page 19 and Chapter 4, “Configuring the File System.”

sync_meta Command

The `sync_meta` command determines whether metadata is written to disk every time it changes. If you are using this command on Sun QFS shared file system, also see “Specifying the Frequency With Which Metadata Is Written: (`sync_meta` Option)” on page 169.

```
:sync_meta eq value
```

For `eq`, specify the equipment number of the file system.

For `value`, specify either 0 or 1, as follows:

- For `value` is 0, metadata is held in a buffer after it changes. For an unshared Sun QFS file system in which higher performance is desired, you can set `value` to 0. In this case, the system performs a delayed write operation in which metadata is held in a buffer before it is written to disk. This value is the default for unshared file systems and for file systems that are not mounted as multireader file systems.
- For `value` is 1, metadata is written to disk every time it changes. This behavior slows performance, but it increases data consistency. This value is the default for Sun QFS file systems mounted as multireader file systems or as shared file systems. For a Sun QFS shared file system, `value` must be set to 1 if failover capability is required.

trace and notrace Commands

The `trace` command enables tracing for a file system. The `notrace` command disables tracing. These global directives affect all operations. For more information about file system tracing, see “`defaults.conf(4)`” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

```
:trace eq
:notrace eq
```

For `eq`, specify the equipment number of a file system.

Miscellaneous Commands

The commands described in this section enable you to control tracing, open access to a disk device, and perform several other miscellaneous tasks.

clear vsn Command

The `clear` command clears the specified VSN from the removable media mount requests display. For more information, see “(p) - Removable Media Load Requests Display” on [page 197](#).

```
:clear vsn
:clear vsn index
```

For *vsn*, specify the volume to mount. Any process waiting for the VSN mount is aborted.

For *index*, specify the decimal ordinal of the VSN in the removable media display.

devlog Command

The `devlog` command sets one or more events to be logged.

```
:devlog eq
:devlog eq option
```

For *eq*, specify the equipment number of a device.

For *option*, specify one or more event types. Possible event types are as follows: `all`, `date`, `default`, `detail`, `err`, `event`, `label`, `mig`, `module`, `msg`, `none`, `retry`, `stage`, `syserr`, and `time`. For information about these options, see “[defaults.conf\(4\)](#)” in *Sun QFS and Sun Storage Archive Manager 5.3 Reference Manual*.

If no option is specified, the system does not change the current events being logged for the *eq* specified.

diskvols flag Command

The `diskvols` command sets or clears flags in the disk volume dictionary.

```
:diskvols volume +flag
:diskvols volume -flag
```

For *volume*, specify the volume in the disk volume dictionary.

For *flag*, specify one of the five flags in the `samu D` display. For information about the disk volume dictionary and the flags, see the `samu(1M)` man page.

dtrace Commands

The `dtrace` commands control the DTrace feature for one or more processes. The DTrace commands specify various tracing options.

```
:dtrace daemon-name on
:dtrace daemon-name off
:dtrace daemon-name.variable value
```

For *daemon-name*, specify the keyword `all` to affect all processes or a process name. If one of the following process names is specified, the tracing command affects that process only: `sam-archiverd`, `sam-catserverd`, `sam-fsd`, `sam-rftd`, `sam-recycler`, `sam-sharefsd`, and `sam-stagerd`.

For *variable* and *value*, specify one of the following *variable* and *value* pairs. The `defaults.conf(4)` man page contains comprehensive information about these arguments.

- *filevalue*
Specify the name of a file to which trace files can be written. This can be a full path name.
- *optionsvalue*
Specify a space-separated list of trace options.
- *agevalue*
Specify the trace file rotation age.

Note – Do not set an age of two minutes or less. If you do, the rotation never happens.

- *sizevalue*
Specify the size of the trace file at which rotation begins.

fs Command

The `fs` command sets the file system to be displayed through the `N` display.

```
:fs fsname
```

For *fsname*, specify the name of the file system to be examined.

mount Command

The `mount` command selects a Sun QFS file system.

```
:mount mount-point
```

For *mount-point*, specify the mount point of a file system.

open Command

The open command enables access to the specified disk device. You must issue this command before you can use the read command, disk sector display (S), or file label display (F).

:open *eq*

For *eq*, specify the equipment number of a device.

read Command

The read command reads the specified sector from the currently opened disk device. You must open the device before it can be read.

:read *addr*

For *addr*, specify the hexadecimal sector address.

refresh Command

The refresh command sets the amount of time between samu screen refreshes.

:refresh *i*

For *i*, specify a time in seconds.

snap Command

The snap command sends a snapshot of a display window to file. The default file is snapshots in the current working directory. To aid in problem reporting, take a snapshot of all of the samu utility's displays. Each new snapshot is appended to the snapshots file. The file can be printed, examined using an editor, or faxed to Oracle customer support staff.

:snap
:snap *filename*

For *filename*, specify the path of a file to receive the display information.

!shell-command Command

The ! command enables you to run a shell command without leaving the samu operator utility.

:! shell-command

For *shell-command*, specify a command.