



Sun StorageTek™ QFS File System Configuration and Administration Guide

Version 4, Update 6

Sun Microsystems, Inc.
www.sun.com

Part No. 819-7935-10
April 2007, Revision A

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

Copyright 2007 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries.

This document and the product to which it pertains are distributed under licenses restricting their use, copying, distribution, and decompilation. No part of the product or of this document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, AnswerBook2, docs.sun.com, Solaris, Java, Solstice DiskSuite, SunPlex, and Sun StorageTek are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and in other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and in other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

Mozilla is a trademark or registered trademark of Netscape Communications Corporation in the United States and other countries.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

U.S. Government Rights—Commercial use. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2007 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, Californie 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. a les droits de propriété intellectuelle relatants à la technologie qui est décrit dans ce document. En particulier, et sans la limitation, ces droits de propriété intellectuelle peuvent inclure un ou plus des brevets américains énumérés à <http://www.sun.com/patents> et un ou les brevets plus supplémentaires ou les applications de brevet en attente dans les Etats-Unis et dans les autres pays.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a.

Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, AnswerBook2, docs.sun.com, Solaris, Java, Solstice DiskSuite, SunPlex, et Sun StorageTek sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

Mozilla est une marque de Netscape Communications Corporation aux Etats-Unis et à d'autres pays.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciées de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

LA DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.



Please
Recycle



Adobe PostScript

Contents

Preface xxvii

1. File System Overview 1

File System Features 1

Volume Management 2

Support for Paged and Direct I/O 2

High Capacity 3

Fast File System Recovery 3

Metadata Storage 4

vnode Interface 4

Shared File System Support 4

Linux Client Support 5

Additional File System Features 6

Design Basics 7

Inode Files and File Characteristics 7

Specifying Disk Allocation Units 8

DAU Settings and File System Geometry 8

ms and maSun StorageTek SAM File Systems 9

Dual and Single Allocation Schemes 9

Data Alignment 11

Stripe Widths on Data Disks	11
File Allocation Methods	14
Metadata Allocation	14
Round-Robin Allocation	15
Striped Allocation	17
Striped Groups	20
Mismatched Striped Groups	23
Example of a Mismatched Striped Group	24
Per-logical unit number (LUN) Allocation Control	28
2. Configuring the File System	29
Using the File System Manager Software	29
▼ To Invoke File System Manager for the First Time	30
Creating Additional Administrator and User Accounts	30
▼ To Create Additional Accounts	31
Assigning Privilege Levels	31
Creating an Account for Multiple Users	32
▼ To Add an Additional Server for File System Manager Access	33
Setting the Session Timeout	34
Function of the <code>mcf</code> File	34
The Equipment Identifier Field	35
The Equipment Ordinal Field	36
The Equipment Type Field	36
The Family Set Field	37
The Device State Field	38
The Additional Parameters Field	38
Examples of <code>mcf</code> Files	38
Interactions Among File Settings, Options, and Directives	40
Initializing a File System	41

Configuration Examples	42
▼ To Create a Sun StorageTek QFS Round-Robin Disk Configuration	42
▼ To Create a Sun StorageTek QFS Striped Disk Configuration	44
▼ To Create a Sun StorageTek QFS Striped Group Configuration	45
3. Performing Operations	49
Viewing Files and File Attributes	49
File Attributes and File States	49
Displaying File Information	51
About the <code>sls(1)</code> Output	51
About the Retention Line	54
Propagating Configuration File Changes to the System	54
▼ To Change <code>mcf</code> or <code>defaults.conf</code> Information in a Sun StorageTek QFS Sun Cluster Environment	55
▼ To Change <code>mcf</code> or <code>defaults.conf</code> File System Information in a SAM-QFS Environment	55
▼ To Change <code>mcf</code> or <code>defaults.conf</code> Removable Media Drive Information	56
Changing the Shared Hosts File	58
▼ To Add New Entries or Change Existing Entries	58
▼ To Change Host Names, Reorder Entries, or Insert Entries	59
Setting Up Mount Parameters	60
The <code>mount(1M)</code> Command	61
The <code>/etc/vfstab</code> File	61
The <code>samfs.cmd</code> File	62
Unmounting a File System	63
▼ To Unmount a Stand-alone QFS or SAM-QFS File System	64
▼ To Unmount a Sun StorageTek QFS Shared File System	64
Adding Disk Cache to a File System	64
▼ To Add Disk Cache to a File System	65

Re-creating a File System 66

- ▼ To Back Up and Re-create a File System 66

4. Configuring a Sun StorageTek QFS Shared File System 69

Mounting and Unmounting Sun StorageTek QFS Shared File Systems 69

- ▼ To Mount a Shared File System 70
- ▼ To Unmount a Shared File System 70

Converting an Unshared File System to a Shared File System 71

- ▼ To Perform a Conversion on the Metadata Server 71
- ▼ To Perform a Conversion on Each Client 73

Converting a Shared File System to an Unshared File System 75

- ▼ To Perform a Conversion on Each Client 75
- ▼ To Perform a Conversion on the Server 76

Adding or Removing a Client Host 77

- ▼ To Add a Client Host 77
- ▼ To Remove a Client Host 82

Updating the `mcf` file in a Sun StorageTek QFS Shared Environment 84

Creating the Local Hosts Configuration File 87

Changing the Metadata Server in a Sun StorageTek QFS Environment 90

- ▼ To Change the Metadata Server When the Metadata Server Is Available 91
- ▼ To Change the Metadata Server When the Metadata Server Is Unavailable 91

Changing the Metadata Server in a SAM-QFS Environment 93

- ▼ To Change the Metadata Server in a SAM-QFS Environment 93

Client-Server Communications in a Sun StorageTek QFS Shared File System 94

5. Administering File System Quotas 97

Overview 97

Types of Quotas, Quota Files, and Quota Records 98

Soft Limits and Hard Limits	99
Disk Blocks and File Limits	99
Enabling Quotas	100
Guidelines for Setting Up Quotas	100
▼ To Configure a New File System to Use Quotas	100
▼ To Configure an Existing File System to Use Quotas	102
▼ To Assign Admin Set IDs to Directories and Files	104
Setting Infinite Quotas	105
▼ To Set an Infinite Quota	105
Enabling Default Quota Values	106
▼ To Enable Default Quota Values for Users, Groups, or Admin Sets	106
Enabling Limits	107
▼ To Enable Limits for Users, Groups, or Admin Sets	107
▼ To Enable or Change Limits for Users, Groups, or Admin Sets Using an Existing Quota File	107
Checking Quotas	109
▼ To Check for Exceeded Quotas	109
Changing and Removing Quotas	111
▼ To Change the Grace Period	111
Changing the Grace Period Expiration	113
▼ To Inhibit Additional File System Resource Allocations	115
▼ To Remove a File System's Quotas	117
▼ To Correct Quotas	118
6. Configuring Sun StorageTek QFS in a Sun Cluster Environment	121
Before You Begin	122
Restrictions	123
How the Sun Cluster System and the Sun StorageTek QFS Software Interact	124
Data Access With a Shared File System	124

Data Access With an Unshared File System	125
Sun StorageTek QFS Support for Solaris Volume Manager for Sun Cluster	125
▼ To Configure a File System With Solaris Volume Manager for Sun Cluster	126
About Configuration Examples	132
Configuring a Sun StorageTek QFS Shared File System in a Sun Cluster Environment	134
Metadata Server Resource Considerations	134
Example Configuration	135
▼ To Prepare to Create Sun StorageTek QFS Shared File Systems	136
▼ To Create the Sun StorageTek QFS Shared File System and Configure Sun Cluster Nodes	139
▼ To Validate the Configuration	140
▼ To Configure the Sun Cluster Data Service for Oracle Real Application Clusters	142
Configuring an Unshared File System in a Sun Cluster Environment	144
Example 1: HA-NFS on Raw Global Devices	145
▼ To Prepare to Create an Unshared Sun StorageTek QFS File System	146
▼ To Create the Sun StorageTek QFS File System and Configure Sun Cluster Nodes	147
▼ To Configure the Network Name Service and the IPMP Validation Testing	149
▼ To Configure HA-NFS and the Sun StorageTek QFS File System for High Availability	151
Example 2: HA-NFS on Volumes Controlled by Solstice DiskSuite/Solaris Volume Manager	153
▼ To Prepare the Solstice DiskSuite/Solaris Volume Manager Software	154
▼ To Prepare For a Sun StorageTek QFS File System	155
▼ To Create the Sun StorageTek QFS File System and Configure Sun Cluster Nodes	156

- ▼ To Configure the Network Name Service and the IPMP Validation Testing 158
- ▼ To Configure HA-NFS and the Sun StorageTek QFS File System for High Availability 158

Example 3: HA-NFS on VxVM Volumes 160

- ▼ To Configure the VxVM Software 161
- ▼ To Prepare to Create a Sun StorageTek QFS File System 162
- ▼ To Create the Sun StorageTek QFS File System and Configure Sun Cluster Nodes 163
- ▼ To Validate the Configuration 164
- ▼ To Configure the Network Name Service and the IPMP Validation Testing 164
- ▼ To Configure HA-NFS and the Sun StorageTek QFS File System for High Availability 164

Configuring Shared Clients Outside the Cluster 165

Configuration Prerequisites 165

Sun StorageTek QFS Metadata Server Sun Cluster Nodes 165

Sun StorageTek QFS Metadata Client Nodes 166

Sun Cluster Device Configuration 166

Requirements for Configuring Clients Outside the Cluster 166

Minimum Software Release Levels 167

Hardware Architecture Supported 167

Storage Requirements 167

Configuration Instructions 167

- ▼ To Edit the `/etc/hosts` File 167
- ▼ To Configure the Metadata Server Network 168
- ▼ To Configure the Metadata Client Network 169
- ▼ To Install and Configure Sun Cluster 169
- ▼ To Configure the Sun StorageTek QFS Metadata Server 170
- ▼ To Configure Sun StorageTek QFS Metadata Client 174

Changing the Sun StorageTek QFS Configuration	177
▼ To Change the Shared File System Configuration	177
▼ To Disable HA-NFS on a File System That Uses Raw Global Devices	178
▼ To Disable HA-NFS on a File System That Uses Solaris Volume Manager- Controlled Volumes	179
▼ To Disable HA-NFS on a Sun StorageTek QFS File System That Uses VxVM-Controlled Volumes	181
High-Availability Sun StorageTek SAM Configuration Using Sun Cluster	183

7. Advanced Topics 185

Using Daemons, Processes, and Tracing	185
Daemons and Processes	186
Trace Files	187
Trace File Content	187
Trace File Rotation	188
Determining Which Processes Are Being Traced	188
Using the <code>setfa(1)</code> Command to Set File Attributes	190
Selecting File Attributes for Files and Directories	190
Preallocating File Space	190
Selecting a File Allocation Method and Stripe Width	191
Selecting a Striped Group Device	192
Configuring WORM-FS File Systems	192
Enabling the WORM-FS Feature	193
WORM “Lite” Options	194
Creating WORM Files	195
Example 1. WORM trigger is <code>chmod 4000</code>	196
Example 2. WORM trigger is <code>chmod 4000</code>	197
Example 3. WORM trigger is <code>chmod -w</code>	198
Example 4. WORM trigger is <code>chmod -w</code>	199
Retention Periods	200

Setting the Default Retention Period	202
Setting the Retention Period Using <code>touch</code>	203
Extending a File's Retention Period	205
Using <code>sls</code> to View WORM-FS Files	205
Using <code>sfind</code> to Find WORM-FS Files	206
Accommodating Large Files	207
Configuring a Multireader File System	208
Using the SAN-QFS File System in a Heterogeneous Computing Environment	210
Before You Begin	212
Enabling the SAN-QFS File System	212
▼ To Enable the SAN-QFS File System on the Metadata Controller	212
▼ To Enable the SAN-QFS File System on the Clients	213
▼ To Install the SANergy Software on the Clients	214
Unmounting the SAN-QFS File System	215
▼ To Unmount the SAN-QFS File System on the SANergy Clients	215
▼ To Unmount the SAN-QFS File System on the Metadata Controller	216
▼ To Unmount the SAN-QFS File System on the Sun StorageTek QFS Clients	216
▼ To Unmount the SAN-QFS File System on the Sun StorageTek QFS Server	216
Troubleshooting: Unmounting a SAN-QFS File System With SANergy File Holds	217
▼ To Unmount a File System in the Presence of SANergy File Holds	217
Block Quotas in a SAN-QFS File System	217
File Data and File Attributes in a SAN-QFS File System	217
Using <code>samgrowfs(1M)</code> to Expand SAN-QFS File Systems	218
SAN-QFS Shared File System and Sun StorageTek QFS Shared File System Comparison	218

Understanding I/O Types	219
Paged I/O	219
Direct I/O	219
I/O Switching	220
Increasing File Transfer Performance for Large Files	220
▼ To Increase File Transfer Performance	220
Enabling Qwrite Capability	223
Setting the Write Throttle	224
Setting the Flush-Behind Rate	225
Tuning the Number of Inodes and the Inode Hash Table	226
The <code>ninodes</code> Parameter	226
The <code>nhino</code> Parameter	227
When to Set the <code>ninodes</code> and <code>nhino</code> Parameters	227
A. Troubleshooting Sun StorageTek QFS	229
Checking File System Integrity and Repairing File Systems	229
▼ To Check a File System	230
▼ To Repair a File System	231
Troubleshooting a Failed or Hung <code>sammkfs(1M)</code> or <code>mount(1M)</code> Command in a Shared File System	231
Recovering From a Failed <code>sammkfs(1M)</code> Command	232
▼ To Verify the <code>mcf</code> File and Propagate <code>mcf</code> File Changes to the System	232
Recovering From a Failed <code>mount(1M)</code> Command	232
▼ To Verify that the File System Can Be Mounted	233
▼ To Use the <code>samfsinfo(1M)</code> and <code>samsharefs(1M)</code> Commands	234
▼ To Use the <code>samfsconfig(1M)</code> Command	236
Recovering From a Hung <code>mount(1M)</code> Command	238
▼ To Verify Network Connections	238
▼ To Verify That the Client Can Reach the Server	240

▼	To Verify That the Server Can Reach the Client	243
▼	To Examine the <code>sam-sharefsd</code> Trace Log	244
	Troubleshooting the Linux Client	246
	Troubleshooting Tools	247
	Frequently Asked Questions	248
B.	Mount Options in a Sun StorageTek QFS Shared File System	251
	Mounting in the Background: the <code>bg</code> Option	252
	Reattempting a File System Mount: the <code>retry</code> Option	252
	Declaring a Sun StorageTek QFS Shared File System: the <code>shared</code> Option	253
	Tuning Allocation Sizes: the <code>minallocsz=<i>n</i></code> and <code>maxallocsz=<i>n</i></code> Options	253
	Using Leases in a Sun StorageTek QFS Shared File System: the <code>rdlease=<i>n</i></code> , <code>wrlease=<i>n</i></code> , and <code>aplease=<i>n</i></code> Options	254
	Enabling Multiple Host Reads and Writes: the <code>mh_write</code> Option	255
	Setting the Minimum Number of Concurrent Threads: the <code>min_pool=<i>n</i></code> Option	256
	Retaining Cached Attributes: the <code>meta_timeo=<i>n</i></code> Option	257
	Specifying Striped Allocation: the <code>stripe</code> Option	257
	Specifying the Frequency With Which Metadata Is Written: the <code>sync_meta=<i>n</i></code> Option	258
	Enabling WORM Functionality: the <code>worm_capable</code> and <code>def_retention</code> Options	258
C.	Using the <code>samu(1M)</code> Operator Utility	259
	Overview	259
	▼ To Invoke <code>samu(1M)</code>	260
	▼ To Display a <code>samu(1M)</code> Screen	260
	▼ To Stop <code>samu(1M)</code>	261
	Interacting With <code>samu(1M)</code>	261
	Entering a Device	261
	Getting Online Help	262

▼ To Access Online Help From a Display Screen 262

Operator Displays 262

(a) - Archiver Status Display 263

Navigation 264

Sample Display 264

Field Descriptions 265

(c) - Device Configuration Display 265

Navigation 266

Sample Display 266

Field Descriptions 267

(C) - Memory Display 267

Sample Display 268

(d) - Daemon Trace Controls Display 268

Sample Display 269

(D) - Disk Volume Dictionary 270

Sample Display 270

Flags 270

(f) - File Systems Display 271

Sample Display 271

Field Descriptions 272

(F) - Optical Disk Label Display 273

(h) - Help Display 273

Navigation 273

Sample Display 274

(I) - Inode Display 274

Navigation 275

Sample Display 275

(J) - Preview Shared Memory Display 276

Navigation	276
Sample Display	277
(K) - Kernel Statistics Display	277
Navigation	277
Sample Display	278
(1) - Usage Display	278
Sample Display	278
(L) - Shared Memory Tables	279
Sample Display	279
(m) - Mass Storage Status Display	280
Sample Display	280
Field Descriptions	281
(M) - Shared Memory Display	282
Navigation	282
Sample Display	283
(n) - Staging Status Display	283
Sample Display	284
(N) - File System Parameters Display	284
Navigation	284
Sample Display	285
(o) - Optical Disk Status Display	286
Navigation	286
Sample Display	286
Field Descriptions	287
(p) - Removable Media Load Requests Display	287
Navigation	288
Sample Display	288
Field Descriptions	289

- Flags 289
- (P) - Active Services Display 290
 - Navigation 290
 - Sample Display 290
- (r) - Removable Media Status Display 290
 - Sample Display 291
 - Field Descriptions 291
- (R) - Sun SAM-Remote Information Display 292
- (s) - Device Status Display 292
 - Navigation 293
 - Sample Display 293
 - Field Descriptions 294
- (S) - Sector Data Display 294
 - Navigation 294
- (t) - Tape Drive Status Display 295
 - Navigation 295
 - Sample Display 295
 - Field Descriptions 296
- (T) - SCSI Sense Data Display 296
 - Navigation 297
- (u) - Staging Queue Display 297
 - Navigation 297
 - Sample Display 298
 - Field Descriptions 298
- (U) - Device Table Display 299
 - Navigation 299
 - Sample Display 300
- (v) - Automated Library Catalog Display 300

Navigation	301
Sample Display	302
Field Descriptions	303
Flags	303
(w) - Pending Stage Queue	304
Navigation	305
Sample Display	305
Field Descriptions	306
Operator Display Status Codes	306
Removable Media Device Display Status Codes	306
File System Display Status Codes	307
Operator Display Device States	308
▼ To Change a Drive State from down to on	309
▼ To Change a Drive State from on to down	310
Operator Commands	311
Device Commands	311
File System Commands: I/O Management	312
The <code>:flush_behind eq value</code> Command	312
The <code>:force_nfs_async eq</code> and <code>:noforce_nfs_async eq</code> Commands	312
The <code>:readahead eq contig</code> Command	313
The <code>:sw_raid eq</code> and <code>:nosw_raid eq</code> Commands	313
The <code>:writebehind eq contig</code> Command	314
The <code>:wr_throttle eq value</code> Command	314
File System Commands: Direct I/O Management	314
The <code>:dio_rd_form_min eq value</code> and <code>:dio_wr_form_min eq value</code> Commands	315
The <code>:dio_rd_ill_min eq value</code> and <code>:dio_wr_ill_min eq value</code> Commands	315

- The `:dio_rd_consec` *eq value*
and `:dio_wr_consec` *eq value* Commands 315
- The `:dio_szero` *eq* and `:nodio_szero` *eq* Commands 316
- The `:forcedirectio` *eq*
and `:noforcedirectio` *eq* Commands 316
- File System Commands: Sun StorageTek QFS Shared File Systems 316
 - The `:meta_timeo` *eq interval* Command 316
 - The `:mh_write` *eq* and `:nomh_write` *eq* Commands 317
 - The `:minallopsz` *eq value*
and `:maxallopsz` *eq value* Commands 317
 - The `:rdlease` *eq interval*, `:wrlease` *eq interval*,
and `:aplease` *eq interval* Commands 317
- File System Commands: Miscellaneous 317
 - The `:abr` *eq* and `:noabr` *eq* Commands 318
 - The `:dmr` *eq* and `:nodmr` *eq* Commands 318
 - The `:invalid` *eq interval* Command 318
 - The `:mm_stripe` *eq value* Command 318
 - The `:qwrite` *eq*
and `:noqwrite` *eq* Commands 319
 - The `:refresh_at_eof` *eq*
and `:norefresh_at_eof` *eq* Commands 319
 - The `:suid` *eq* and `:nosuid` *eq* Commands 319
 - The `:stripe` *eq value* Command 320
 - The `:sync_meta` *eq value* Command 320
 - The `:trace` *eq* and `:notrace` *eq* Commands 321
- Miscellaneous Commands 321
 - The `:clear` *vsu [index]* Command 321
 - The `:devlog` *eq [option]* Command 321
 - The `:diskvols` *volume [+flag | -flag]* Command 321
 - The `:dtrace` Commands 322
 - The `:fs` *fsname* Command 322

The <code>:mount <i>mntpt</i></code> Command	323
The <code>:open <i>eq</i></code> Command	323
The <code>:read <i>addr</i></code> Command	323
The <code>:refresh <i>i</i></code> Command	323
The <code>:snap [<i>filename</i>]</code> Command	323
The <code>:! <i>shell_command</i></code> Command	323

Glossary	325
-----------------	------------

Index	337
--------------	------------

Figures

FIGURE 1-1	Round-Robin Allocation in an <code>ms</code> File System Using Five Devices	16
FIGURE 1-2	Round-Robin Allocation in an <code>ma</code> File System Using Five Devices	17
FIGURE 1-3	Striping in an <code>ms</code> File System Using Five Devices	19
FIGURE 1-4	Striping in an <code>ma</code> File System Using Five Devices	20
FIGURE 1-5	Sun StorageTek QFS Round-Robin Striped Groups	21
FIGURE 1-6	Sun StorageTek QFS Striped Group Allocation	23
FIGURE 1-7	Sun StorageTek QFS File System Using Mismatched Striped Groups in a Striped Allocation	25
FIGURE 4-1	Network Interfaces	89
FIGURE 7-1	SAN-QFS File System Using Sun StorageTek QFS Software and SANergy Software	211

Tables

TABLE 1-1	Product Overview	1
TABLE 1-2	Content of <code>.inode</code> Files	7
TABLE 1-3	Sun StorageTek QFS Equipment Type Values and DAU Sizes	10
TABLE 1-4	<code>ms</code> File System Default Stripe Widths	12
TABLE 1-5	Default Stripe Widths	13
TABLE 1-6	Default Allocation Methods	14
TABLE 1-7	Metadata Allocation	15
TABLE 1-8	Sample File System Characteristics	26
TABLE 2-1	File System Manager Permission Levels	31
TABLE 2-2	Sun StorageTek QFS or SAM-QFS Equipment Type Field	36
TABLE 3-1	User-Specified File Attributes	50
TABLE 3-2	<code>s1s(1)</code> Output Explanation	51
TABLE 4-1	Local Hosts Configuration File Fields	87
TABLE 5-1	Quota File Names	98
TABLE 7-1	Daemons and Processes	186
TABLE 7-2	File Allocations and Stripe Widths	191
TABLE 7-3	SAN-QFS Shared File System Versus Sun StorageTek QFS Shared File System	218
TABLE A-1	Error Indicators	229
TABLE A-2	<code>/proc</code> files	247
TABLE B-1	Lease-Related <code>mount(1M)</code> Options	254

TABLE B-2	File Access Based on the <code>mh_write</code> Option	255
TABLE C-1	Control Keys for the <code>a</code> Display	264
TABLE C-2	Control Keys for the <code>:a filesystem</code> Display	264
TABLE C-3	<code>samu(1M) a</code> Display Field Descriptions	265
TABLE C-4	Control Keys for the <code>c</code> Display	266
TABLE C-5	<code>samu(1M) c</code> Display Field Descriptions	267
TABLE C-6	Flags Field for the <code>samu(1M) D</code> Display	270
TABLE C-7	<code>samu(1M) f</code> Display Field Descriptions	272
TABLE C-8	Control Keys for the <code>h</code> Display	273
TABLE C-9	Control Keys for the <code>I</code> Display	275
TABLE C-10	Control Keys for the <code>J</code> Display	276
TABLE C-11	Control Keys for the <code>K</code> Display	277
TABLE C-12	<code>samu(1M) m</code> Display Field Descriptions	281
TABLE C-13	Control Keys for the <code>M</code> Display	282
TABLE C-14	Control Keys for the <code>N</code> Display	284
TABLE C-15	Control Keys for the <code>o</code> Display	286
TABLE C-16	<code>samu(1M) o</code> Display Field Descriptions	287
TABLE C-17	Control Keys for the <code>p</code> Display	288
TABLE C-18	<code>samu(1M) p</code> Display Field Descriptions	289
TABLE C-19	Flags Field for the <code>samu(1M) p</code> Display	289
TABLE C-20	Control Keys for the <code>P</code> Display	290
TABLE C-21	<code>samu(1M) r</code> Display Field Descriptions	291
TABLE C-22	Control Keys for the <code>s</code> Display	293
TABLE C-23	<code>samu(1M) s</code> Display Field Descriptions	294
TABLE C-24	Control Keys for the <code>S</code> Display	294
TABLE C-25	Control Keys for the <code>t</code> Display	295
TABLE C-26	<code>samu(1M) t</code> Display Field Descriptions	296
TABLE C-27	Control Keys for the <code>T</code> Display	297
TABLE C-28	Control Keys for the <code>u</code> Display	297
TABLE C-29	<code>samu(1M) u</code> Display Field Descriptions	298

TABLE C-30	Control Keys for the <code>U</code> Display	299
TABLE C-31	Control Keys for the <code>v</code> Display	301
TABLE C-32	<code>samu(1M)</code> <code>v</code> Display Field Descriptions	303
TABLE C-33	Flags Field for <code>samu(1M)</code> <code>v</code> Display	303
TABLE C-34	Control Keys for the <code>w</code> Display	305
TABLE C-35	<code>samu(1M)</code> <code>w</code> Display Field Descriptions	306
TABLE C-36	Removable Media Device Display Status Codes	306
TABLE C-37	File System Display Status Codes	308
TABLE C-38	Operator Display Device States	308
TABLE C-39	Device Command Actions	311
TABLE C-40	Tracing Command Arguments	322

Preface

This manual, the *Sun StorageTek™ QFS File System Configuration and Administration Guide*, describes the file system software included in Sun StorageTek QFS Version 4, Update 6 (4U6).

The Sun StorageTek QFS product includes a file system that can be used as a stand-alone file system, as a shared file system, or in conjunction with the storage and archive manager contained in the Sun StorageTek SAM product. When used in conjunction with Sun StorageTek Storage Archive Manager (SAM) software, it is known as *SAM-QFS*.

The Sun StorageTek QFS software package runs in the following operating system (OS) environments:

- Solaris™ 9 Operating System (OS) 04/03 or later for SPARC® platforms
- Solaris 10 OS for SPARC or x64 platforms
- Solaris 10 OS for x86 (32-bit) (Sun StorageTek QFS shared client only)
- Red Hat Enterprise Linux 3.0 (UD-5 and UD8) for x86/x64 platforms (Sun StorageTek QFS shared client only)
- Red Hat Enterprise Linux 4.0 (UD-2 and UD-4) for x64 platforms (Sun StorageTek QFS shared client only)
- SuSE Linux Enterprise Server 8 (service pack 4) for x64 platforms (Sun StorageTek QFS shared client only)
- SuSE Linux Enterprise Server 9 (service pack 2) for x64 platforms and SGI Altix Itanium systems (Sun StorageTek QFS shared client only)
- SuSE Linux Enterprise Server 10 for x64 platforms (Sun StorageTek QFS shared client only)

Note – This is the last version of Sun StorageTek QFS that will support SuSE Enterprise Linux 8. It will not be supported in the next version of the software.

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

How This Book Is Organized

This manual contains the following chapters:

- [Chapter 1](#) provides overview information.
- [Chapter 2](#) provides file system configuration information.
- [Chapter 3](#) explains how to perform various tasks, such as initializing a file system, adding a server, adding disk cache, and other system administration activities.
- [Chapter 4](#) explains how to configure a Sun StorageTek QFS shared file system.
- [Chapter 5](#) explains how to use file system quotas.
- [Chapter 6](#) describes how to configure Sun StorageTek QFS software in a Sun™ Cluster environment.
- [Chapter 7](#) explains miscellaneous advanced topics such using a multireader file system and performance features.
- [Appendix A](#) contains information about troubleshooting the Sun StorageTek QFS software.
- [Appendix B](#) provides a list of the mount options for the Sun StorageTek QFS shared file system.
- [Appendix C](#) explains how to use the samu(1M) operator utility.

Using UNIX Commands

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

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

<http://docs.sun.com>

Shell Prompts

TABLE P-1 shows the shell prompts used in this manual.

TABLE P-1 Shell Prompts

Shell	Prompt
C shell	<i>machine-name%</i>
C shell superuser	<i>machine-name#</i>
Bourne shell and Korn shell	\$
Bourne shell and Korn shell superuser	#

Typographic Conventions

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

TABLE P-2 Typographic Conventions

Typeface or Symbol	Meaning	Examples
AaBbCc123	The names of commands, files, and directories; on-screen computer output.	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>% You have mail.</code>
AaBbCc123	What you type, when contrasted with on-screen computer output.	<code>% su</code> Password:
<i>AaBbCc123</i>	Book titles; new words or terms; words to be emphasized; and command-line variables to be replaced with a real name or value.	Read Chapter 6 in the <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be root to do this. To delete a file, type <code>rm filename</code> .

TABLE P-2 Typographic Conventions (Continued)

Typeface or Symbol	Meaning	Examples
[]	In syntax, brackets indicate that an argument is optional.	<code>scmadm [-d sec] [-r n[:n][,n]...] [-z]</code>
{ arg arg }	In syntax, braces and pipes indicate that one of the arguments must be specified.	<code>sndradm -b { phost shost }</code>
\	At the end of a command line, the backslash (\) indicates that the command continues on the next line.	<code>atm90 /dev/md/rdisk/d5 \ /dev/md/rdisk/d1 atm89</code>

Related Documentation

This manual is part of a set of documents that describes the operations of the Sun StorageTek QFS and Sun StorageTek SAM software products. [TABLE P-3](#) shows the complete release 4U6 documentation set for these products.

TABLE P-3 Related Sun StorageTek Documentation

Title	Part Number
<i>Sun StorageTek QFS Installation and Upgrade Guide</i>	819-7936-10
<i>Sun StorageTek Storage Archive Manager File System Configuration and Administration Guide</i>	819-7934-108
<i>Sun StorageTek Storage Archive Manager Archive Configuration and Administration Guide</i>	819-7931-10
<i>Sun StorageTek Storage Archive Manager Installation and Upgrade Guide</i>	819-7932-10
<i>Sun StorageTek Storage Archive Manager Troubleshooting Guide</i>	819-7933-10
<i>Sun StorageTek QFS and Sun StorageTek Storage Archive Manager 4.6 Release Notes</i>	818-7938-10
<i>Sun StorageTek QFS Linux Client Guide</i>	819-7937-10

Accessing Sun Documentation Online

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

To Access Documentation From `docs.sun.com`

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

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

`http://docs.sun.com`

The `docs.sun.com` page appears.

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

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

This web site contains documentation for Network Storage products.

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

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

The Storage Software page appears.

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

Third-Party Web Sites

Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites

or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Contacting Sun Technical Support

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

<http://www.sun.com/service/contacting>

Licensing

For information on obtaining licenses for Sun StorageTek QFS and Sun StorageTek SAM software, contact your Sun sales representative or your authorized service provider (ASP).

Installation Assistance

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

Sun Welcomes Your Comments

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

<http://docs.sun.com/app/docs/form/comments>

Please include the title and part number of your document with your feedback (*Sun StorageTek QFS File System Configuration and Administration Guide*, part number 819-7935-10).

File System Overview

This chapter contains the following sections:

- [“File System Features” on page 1](#)
- [“Design Basics” on page 7](#)
- [“File Allocation Methods” on page 14](#)

File System Features

The Sun StorageTek QFS file system is a configurable file system that presents a standard UNIX file system (UFS) interface to users. [TABLE 1-1](#) shows how the file system can be used or combined with the storage and archive management (Sun StorageTek SAM) software.

TABLE 1-1 Product Overview

Product	Components
Sun StorageTek QFS file system	A stand-alone file system.
Sun StorageTek QFS shared file system	A distributed file system that can be mounted on multiple host systems.

TABLE 1-1 Product Overview (*Continued*)

Product	Components
SAM file system	The file system that is included with the Sun StorageTek SAM software. This file system does not include some of the features found in the Sun StorageTek QFS file system.
SAM-QFS	When the Sun StorageTek QFS and the Sun StorageTek SAM software are used together, you can take advantage of the advanced file system features in the Sun StorageTek QFS product as well as the storage management features of the Sun StorageTek SAM product. This combination is called SAM-QFS. Note: Unless otherwise noted, Sun StorageTek QFS file system information throughout this manual applies to SAM-QFS configurations as well.

The Sun StorageTek QFS file system does not require changes to user programs or to the UNIX kernel. Some of the features of the Sun StorageTek QFS file system are described in the following sections.

Volume Management

Sun StorageTek 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 is in contrast to most UNIX file systems, which can address only one device or one portion of a device. Sun StorageTek QFS file systems do not require additional volume management applications. However, if you want to use mirroring for devices in a Sun StorageTek QFS environment, you must obtain an additional package, such as a logical volume manager.

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

Support for Paged and Direct I/O

The Sun StorageTek QFS file system supports two different types of I/O: paged (also called cached or buffered I/O) and direct. These I/O types perform as follows:

- When *paged I/O* is used, user data is cached in virtual memory pages and the kernel writes the data to disk. The standard Solaris OS interfaces manage *paged I/O*. This is the default type of I/O for Sun StorageTek QFS.

- When *direct I/O* is used, user data is written directly from user memory to disk. You can specify direct I/O by using the Solaris OS `directio(3C)` function call or the `setfa(1)` command with its `-D` option. By using *direct I/O*, you can realize substantial performance improvements for large block, sequential, aligned I/O.

High Capacity

The Sun StorageTek QFS software supports files of up to 2^{63} bytes in length. Such very large files can be striped across many disks or RAID devices, even within a single file system. This is true because Sun StorageTek QFS file systems use true 64-bit addressing, in contrast to standard UNIX file systems, which 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. 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 StorageTek 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 512 bytes of storage per file.

In a Sun StorageTek QFS file system, the inodes are located on the metadata devices and are separated from the file data devices. In practice, the size of your metadata (`mm`) devices limits the number of files in a Sun StorageTek 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 10^7 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(1M)`) to repair inconsistencies after a system failure.

A Sun StorageTek 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 `sync(1M)`). In addition, Sun StorageTek QFS file systems recover from system failures without using journaling. They accomplish this dynamically by using identification records, serial writes, and error checking for all critical I/O operations. After a system failure, even multiterabyte-sized Sun StorageTek 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 StorageTek QFS file system has the option of separating the file system metadata from the file data by storing them on separate devices. The Sun StorageTek QFS file system enables you to define one or more separate metadata devices in order to reduce device head movement and rotational latency, improve RAID cache utilization, or mirror metadata without mirroring file data.

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

vnode Interface

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

By using the `vfs/vnode` interface, the file system works with the standard 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 StorageTek QFS file systems. If a file is identified as a Sun StorageTek QFS file, the kernel passes the request to the appropriate file system for handling. Sun StorageTek QFS file systems are identified as type `samfs` in the `/etc/vfstab` file and through the `mount(1M)` command.

Shared File System Support

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

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

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

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

- These 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 `EACCESS` error is returned if the mandatory lock is set. Advisory locks are supported, however. For more information about advisory locks, see the `fcntl(2)` system call.

For more information about shared file systems, see [Chapter 4, “Configuring a Sun StorageTek QFS Shared File System”](#) on page 69.

Linux Client Support

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

- Red Hat Enterprise Linux 3.0 (UD-5 and UD8) for x86/x64 platforms (Sun StorageTek QFS shared client only)
- Red Hat Enterprise Linux 4.0 (UD-2 and UD-4) for x64 platforms (Sun StorageTek QFS shared client only)
- SuSE Linux Enterprise Server 8 (service pack 4) for x64 platforms (Sun StorageTek QFS shared client only)
- SuSE Linux Enterprise Server 9 (service pack 2) for x64 platforms and SGI Altix Itanium systems (Sun StorageTek QFS shared client only)
- SuSE Linux Enterprise Server 10 for x64 platforms (Sun StorageTek QFS shared client only)

Note – This is the last version of Sun StorageTek QFS that will support SuSE Enterprise Linux 8. It will not be supported in the next version of the software.

Unlike shared Sun StorageTek QFS Solaris clients, the Linux client is restricted to client-only behavior. It cannot be configured as a potential metadata server. The Linux client supports interaction with Sun StorageTek SAM software, but does not support Sun StorageTek SAM commands such as `stage`, `archive`, `release`, and `samu`. It has Sun StorageTek QFS file system functionality only.

For more information about the Sun StorageTek QFS Linux client software, see the *Sun StorageTek QFS Linux Client Guide*.

Additional File System Features

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

- **Preallocation of file space** – You can use the `setfa(1)` command to preallocate contiguous disk space for fast sequential reads and writes.
- **Application programming interface (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 the `intro_libsam(3)` man page.
- **Adjustable disk allocation units (DAUs)** –The DAU is the basic unit of online storage. The Sun StorageTek 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. You can adjust the DAU size in multiples of 4 kilobytes. For more information, see [“Specifying Disk Allocation Units” on page 8](#).
- **Support for multiple striped groups** – To support multiple RAID devices in a single file system, Sun StorageTek 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)` command.
- **Sun Cluster system interoperability** – The Sun StorageTek QFS file system is supported as a local file system and as a highly available file system in a Sun Cluster environment. For more information, see [“Configuring Sun StorageTek QFS in a Sun Cluster Environment” on page 121](#).

Design Basics

Sun StorageTek QFS file systems are multithreaded, advanced storage management systems. To take maximum advantage of the software's capabilities, create multiple file systems whenever possible.

Sun StorageTek 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. The DNLC feature is available in all Solaris OS 9 and later releases.

The following sections cover some additional features that affect file system design:

- [“Inode Files and File Characteristics” on page 7](#)
- [“Specifying Disk Allocation Units” on page 8](#)

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. A Sun StorageTek QFS `.inodes` file can reside on a metadata device that is separate from the file data device.

Like a standard Solaris OS inode, a Sun StorageTek QFS file system inode contains the file's POSIX standard inode times: file access, file modification, and inode changed times. A Sun StorageTek QFS file system inode includes other times as well, as shown in [TABLE 1-2](#).

TABLE 1-2 Content of `.inode` Files

Time	Incident
access	Time the file was last accessed. POSIX standard.

TABLE 1-2 Content of `.inode` Files (*Continued*)

Time	Incident
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 Sun StorageTek QFS file system were last changed. Sun Microsystems extension.
creation	Time the file was created. Sun Microsystems extension.
residence	Time the file changed from offline to online or vice versa. Sun Microsystems extension.

Note – If the WORM-FS (write once read many) package is installed, the inode also includes a retention-end date. See [“Configuring WORM-FS File Systems” on page 192](#) for more information.

For more information on viewing inode file information, see [“Viewing Files and File Attributes” on page 49](#).

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.

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

DAU Settings and File System Geometry

Sun StorageTek QFS file systems use an adjustable DAU. You can use this configurable 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 220](#).

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 the `sammkfs(1M)` command 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 [“Configuring the File System” on page 29](#).

ms and maSun StorageTek SAM File Systems

Two file allocation schemes are available to you: an `ms` file system type, and an `ma` file system type.

For a simple Sun StorageTek QFSSun StorageTek SAM 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 StorageTek 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 `gXXX` devices.

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

- `mm` and `mr` devices
- `mm` and `gXXX` devices
- `mm`, `mr`, and `gXXX` devices
- `mm` and `md` devices

For more information on these device types, see Table 2-2.

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 `-a allocation-unit` option to the `sammkfs(1M)` 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 11](#) 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 Topics” on page 185](#).

Only `ma` Sun StorageTek 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 `gXXX`. The `md` devices are limited to DAU sizes of 16 kilobytes, 32 kilobytes, or 64 kilobytes.

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

The DAU size for Sun StorageTek QFS file systems that use `mr` and `gXXX` 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 `mcf` file. [TABLE 1-3](#) shows these DAU sizes.

TABLE 1-3 Sun StorageTek QFS 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 16 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 1-3 Sun StorageTek QFS Equipment Type Values and DAU Sizes (*Continued*)

Equipment Type	DAU Sizes
<code>md</code>	<p>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.</p> <p>An <code>md</code> device in an <code>ma</code> file system is used to store data only, not metadata. An <code>md</code> device in an <code>ms</code> file system is used to store both file data and metadata.</p>

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 may 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 StorageTek QFS file system. To find out whether you are using a version 1 superblock, use the `samfsinfo(1M)` 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 Sun StorageTek QFS 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 (`mx` 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 Sun StorageTek QFS `ms` and `ma` file systems. The stripe width is specified by the `-o stripe=n` option in the `mount(1M)` command. If the stripe width is set to 0, round-robin allocation is used.

The following subsections describe stripe widths on the various file systems.

On `ms` file systems, the stripe width is set at mount time. [TABLE 1-4](#) shows default stripe widths.

TABLE 1-4 `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 `sammkfs(1M)` is run with default settings, the default large DAU is 64 kilobytes. If no stripe width is specified when the `mount(1M)` command is issued, the default is used, and the stripe width set at mount time is 2.

Note – It is important that the stripe width be set to greater than zero in an `ms` file system so that metadata information is striped across the disk.

Note that if you multiply the number in the first column of [TABLE 1-4](#) by the number in the second column, the resulting number is 128 kilobytes. Sun StorageTek 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 Sun StorageTek QFS `ma` file systems, the stripe width that is set at mount time depends on whether or not striped groups are configured. A striped group is a collection of devices that are striped as a group. For more information about striped groups, see [“File Allocation Methods” on page 14](#). This section describes stripe widths for Sun StorageTek 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 StorageTek QFS file systems are most efficient if write operations write at least one whole stripe per I/O request. [TABLE 1-5](#) shows the default stripe widths.

TABLE 1-5 Default Stripe Widths

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
40 kilobytes	3 DAUs	120 kilobytes
48 kilobytes	2 DAUs	96 kilobytes
56 kilobytes	2 DAUs	112 kilobytes
64 kilobytes (default)	2 DAUs	128 kilobytes
72 kilobytes	1 DAU	72 kilobytes
128 kilobytes	1 DAU	128 kilobytes
> 128 kilobytes	1 DAU	DAU size

Stripe Widths on ma File Systems Using Striped Groups

If striped groups are configured for your Sun StorageTek QFS file system, the minimum amount of space allocated is the DAU multiplied by the number of devices in the striped group. The amount of the allocation can be very large 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 on allocation methods, see [“File Allocation Methods” on page 14](#).

Stripe Widths on Metadata Disks

You can use the `-o mm_stripe=n` option to the `mount_samfs(1M)` 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(1M)` 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 StorageTek QFS software enables you to specify both round-robin and striped allocation methods. [TABLE 1-6](#) shows the default file allocation methods used.

TABLE 1-6 Default Allocation Methods

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

The rest of this section describes allocation in more detail.

Metadata Allocation

Metadata allocation varies according to the type of file system you have:

- 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. TABLE 1-7 shows how the system allocates metadata.

TABLE 1-7 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-, 32-, or 64-kilobyte DAU
Indirect blocks	16-kilobyte DAU	16-, 32-, or 64-kilobyte DAU
Directories	4-kilobyte blocks and 16-kilobyte DAUs	4 kilobytes, up to a 32-kilobyte total, then DAU size

Round-Robin Allocation

The round-robin allocation method writes one data file at a time to each successive device in the family set. Round-robin allocation is useful for multiple data streams, because 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.

You can specify round-robin allocation explicitly in the */etc/vfstab* file by entering `stripe=0`.

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

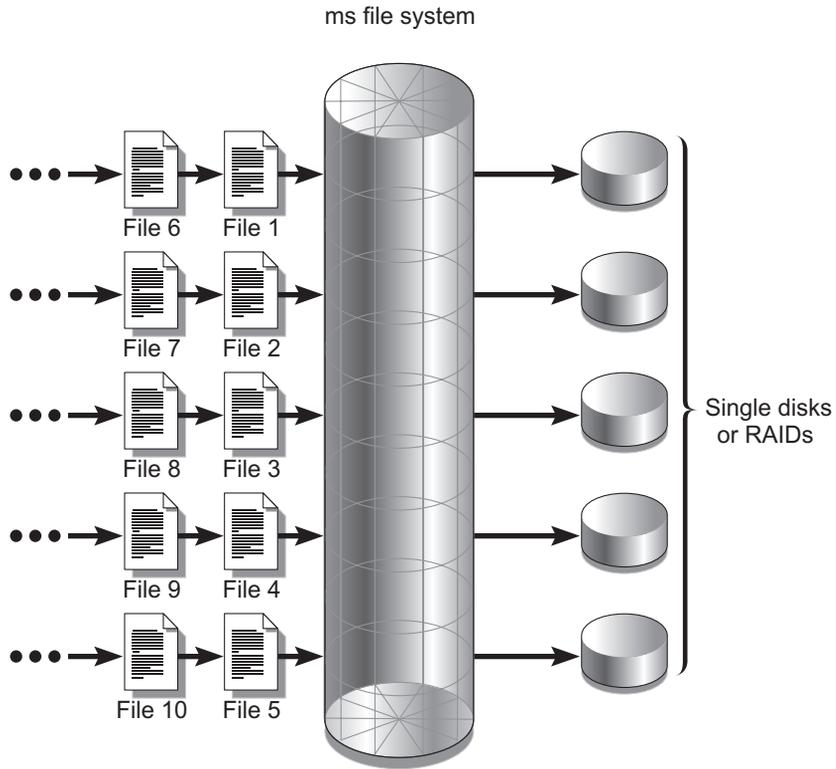


FIGURE 1-1 Round-Robin Allocation in an ms File System Using Five Devices

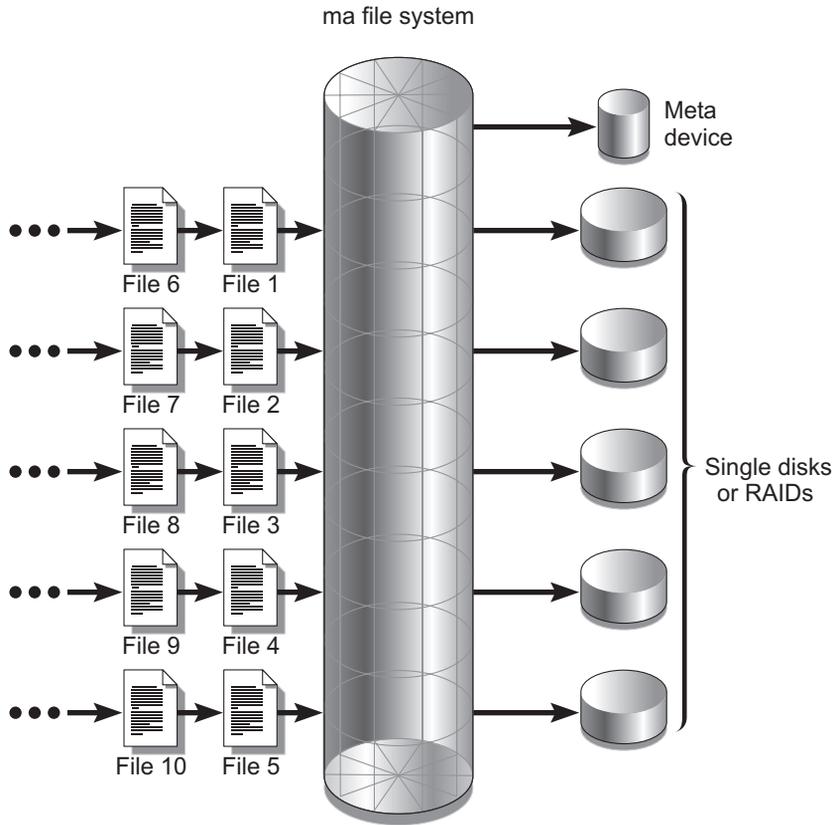


FIGURE 1-2 Round-Robin Allocation in an ma File System Using Five Devices

Striped Allocation

By default, Sun StorageTek 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 is using striped devices addresses blocks in an interlaced fashion rather than sequentially. Striping generally increases performance because it enables multiple I/O streams to simultaneously write a file across multiple disks. The DAU and the stripe width determine the size of the I/O transmission.

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

When a Sun StorageTek QFS 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 Sun StorageTek QFS 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 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 using striped allocations. In these figures, $DAU \times \textit{stripe-width}$ bytes of the file are written to disk 1. $DAU \times \textit{stripe-width}$ 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.

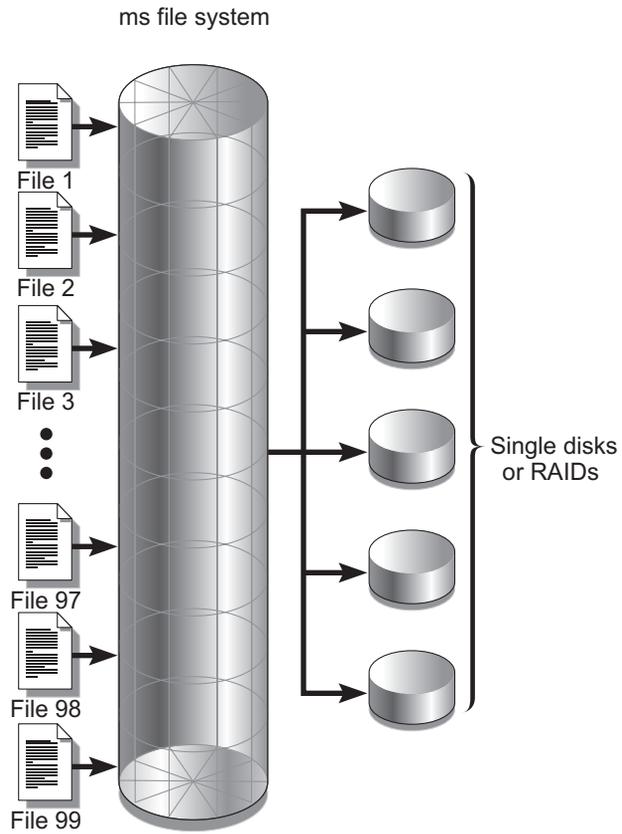


FIGURE 1-3 Striping in an ms File System Using Five Devices

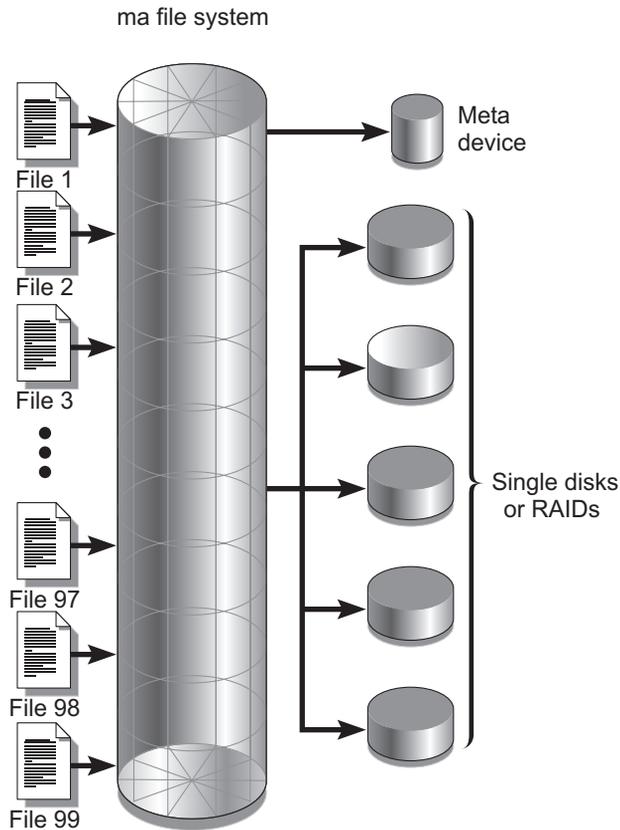


FIGURE 1-4 Striping in an ma File System Using Five Devices

Striped Groups

A striped group is a Sun StorageTek QFS allocation method designed for file systems that have extremely large I/O requirements and terabytes of disk cache. A striped group enables you to designate an Equipment Type value that accounts for multiple physical disks. Multiple striped group Equipment Type entries can make up a single Sun StorageTek 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 StorageTek QFS file system. Defined in the `mcf` file as `gXXX` devices, striped groups enable one file to be written to and read from two or more devices. You can specify up to 128 striped groups within a file system.

FIGURE 1-5 depicts a Sun StorageTek QFS ma file system using striped groups and a round-robin allocation. In FIGURE 1-5, 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.

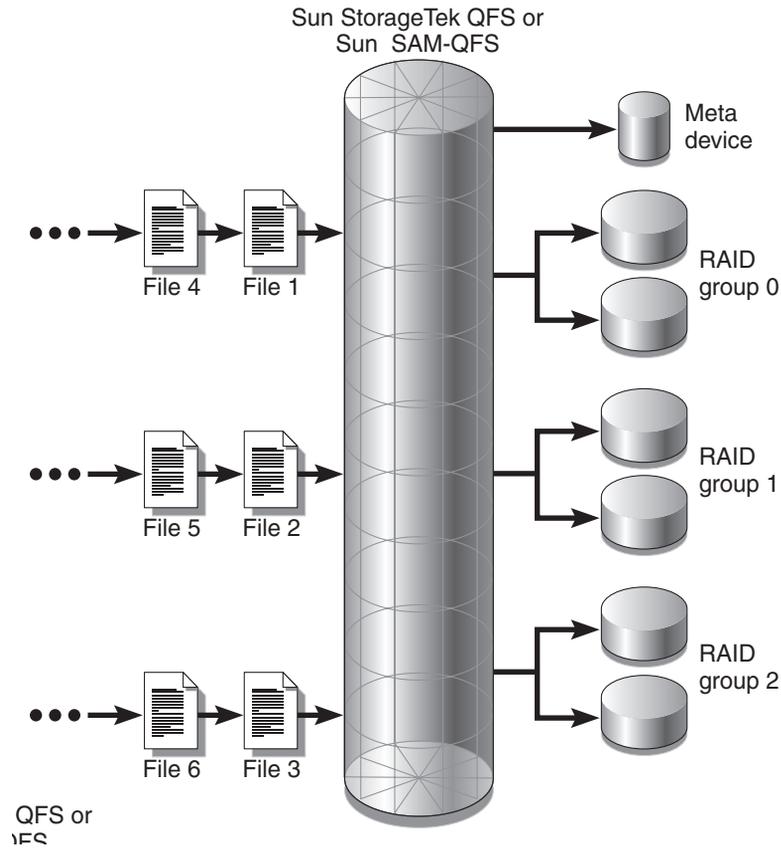


FIGURE 1-5 Sun StorageTek QFS Round-Robin Striped Groups

For the configuration in [FIGURE 1-5](#), the mount point option in `/etc/vfstab` is set to `stripe=0`. [CODE EXAMPLE 1-1](#) shows the `mcf` file that declares these striped groups.

CODE EXAMPLE 1-1 Example `mcf` File Showing Striped Groups

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

[FIGURE 1-6](#) depicts a Sun StorageTek 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.

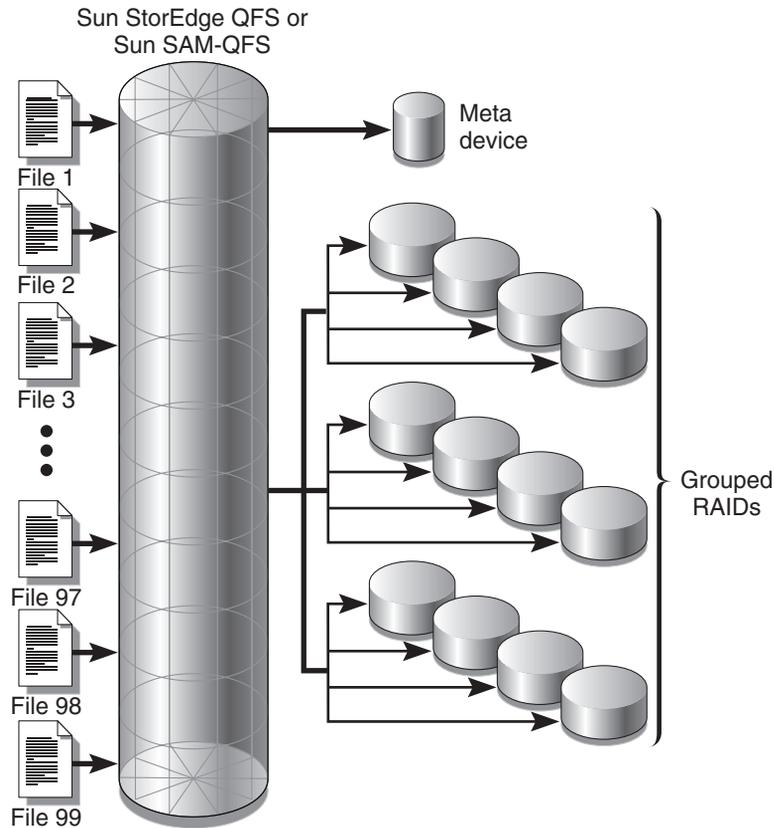


FIGURE 1-6 Sun StorageTek QFS Striped Group Allocation

Mismatched Striped Groups

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

Note – 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(1)` command's `-g` option to direct files into the desired group. For more information, see the `setfa(1)` man page.

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

The following example shows how a file system can be set up 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.

[FIGURE 1-7](#) depicts the file system needed. It is an `ma` file system using mismatched striped groups in a striped allocation.

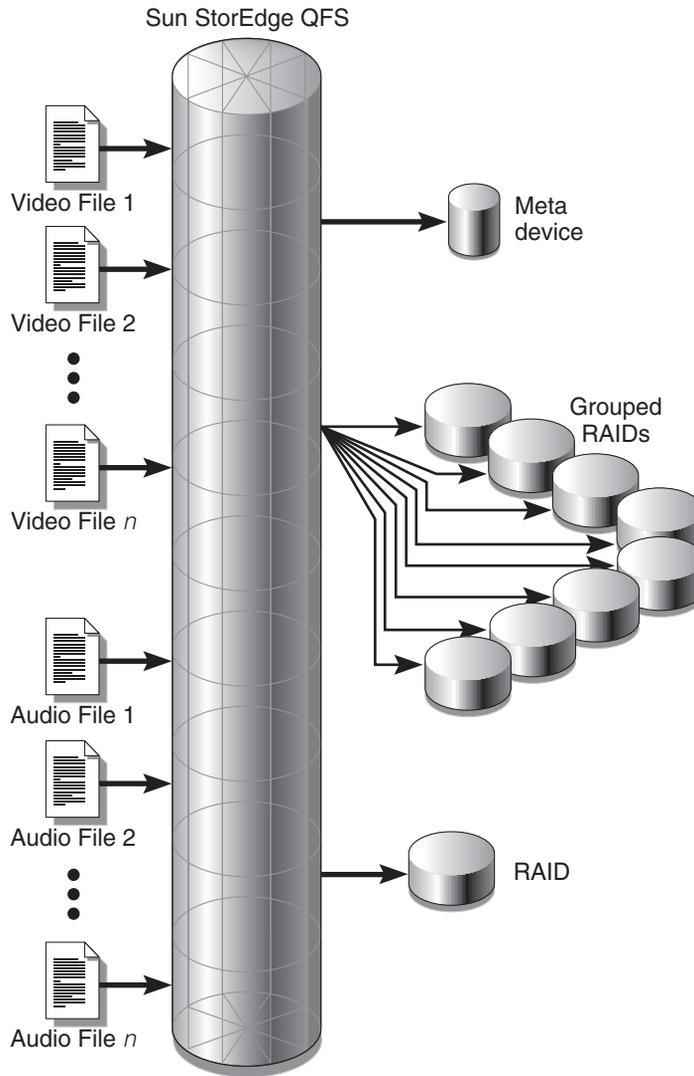


FIGURE 1-7 Sun StorageTek QFS File System Using Mismatched Striped Groups in a Striped Allocation

TABLE 1-8 shows the characteristics of this sample file system.

TABLE 1-8 Sample File System Characteristics

Characteristics	Notes
File system name	avfs
Number of stripe groups	Two. The video file group is g0. The audio file group is g1.
Stripe width	0
DAU	128 kilobytes
Number of disks for g0	Eight
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 is used because a value greater than 0 is not supported for mismatched striped groups.

CODE EXAMPLE 1-2 shows the `mcf` file for file system `avfs`.

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

```
# Equipment      Eq  Eq  Fam  Dev  Additional
# Identifier      Ord Type Set  State Parameters
#
avfs              100 ma  avfs
/dev/dsk/c00t1d0s6 101 mm  avfs -
#
/dev/dsk/c01t0d0s6 102 g0  avfs -
/dev/dsk/c02t0d0s6 103 g0  avfs -
/dev/dsk/c03t0d0s6 104 g0  avfs -
/dev/dsk/c04t0d0s6 105 g0  avfs -
```

CODE EXAMPLE 1-2 The `mcf` File for File System `avfs` (Continued)

```
/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 enter the `sammkfs(1M)` and `mount(1M)` commands shown in [CODE EXAMPLE 1-3](#) to create and mount the `avfs` file system.

CODE 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 [CODE EXAMPLE 1-4](#) to create two directories for the two types of files.

CODE 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(1)` commands shown in [CODE EXAMPLE 1-5](#) 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.

CODE EXAMPLE 1-5 Commands to Set File Attributes

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

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

Per-logical unit number (LUN) Allocation Control

If necessary, you can disable allocation to a specific Sun StorageTek QFS data partition by using a `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 [“Using the samu\(1M\) Operator Utility” on page 259](#).

Configuring the File System

The installation and configuration process is described completely in the *Sun StorageTek QFS Installation and Upgrade Guide*. This chapter provides additional information about configuring the file system used in the Sun StorageTek QFS environment. This chapter contains the following sections:

- [“Using the File System Manager Software” on page 29](#)
 - [“Function of the mcf File” on page 34](#)
 - [“Initializing a File System” on page 41](#)
 - [“Configuration Examples” on page 42](#)
-

Using the File System Manager Software

The File System Manager software is a browser interface tool that enables you to configure, control, protect, and monitor one or more file systems in your network from a central location. To access this central location, you can use the web browser on any host in your network.

The goal of the software is to provide a less complex way than command-line interface (CLI) commands of performing the most common tasks associated with file systems. For instructions on installing the File System Manager software, see the *Sun StorageTek QFS Installation and Upgrade Guide*.

By default, File System Manager is set up to manage the server on which it is installed. It can also be used to manage other servers running Sun StorageTek QFS software, but those additional servers must first be configured to allow File System Manager access. For instructions on adding additional managed servers, see [“To Add an Additional Server for File System Manager Access” on page 33](#).

▼ To Invoke File System Manager for the First Time

Perform this procedure if you want to invoke File System Manager and use it, rather than CLI commands, to perform file system administration and configuration.

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

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

For *hostname*, type the name of the host where the File System Manager software is installed. If you need to specify a domain name in addition to the host name, specify the *hostname* in this format: *hostname.domainname*. Note that this URL begins with `https`, not `http`.

The Sun Java™ Web Console login page is displayed.

4. **At the User Name prompt, enter `root` or another valid user name.**

Note – If you have upgraded the File System Manager software from an earlier version, the `samadmin` user account is also available. You may type `samadmin` in the User Name field and then type the `samadmin` password to gain full access to all File System Manager operations.

5. **At the Password prompt, enter the password.**
6. **Click Log In.**
7. **In the Storage section, click File System Manager.**

You are now logged in to File System Manager.

Creating Additional Administrator and User Accounts

You can create additional administrator and guest accounts at any time after the initial File System Manager configuration. These guest accounts are local to the management station.

If you remove the File System Manager software, the removal scripts do not remove any additional accounts that you create manually. It is your responsibility to use one or both of the following procedures to administer any accounts you add manually.

▼ To Create Additional Accounts

1. **Outside of the browser interface, log in to the management station server as root.**
2. **Use the `useradd` and `passwd` commands to add each user.**

For example, to add a user with account name `bobsmith`, type the following:

```
# /usr/sbin/useradd bobsmith
# /usr/bin/passwd bobsmith
```

Each user account that you add in this way has read-only viewing privileges for File System Manager functions. To add additional privileges see the following section, [“Assigning Privilege Levels” on page 31](#).

Assigning Privilege Levels

You can assign users full or partial access to File System Manager functions. The following table lists the five levels of privileges you can assign to File System Manager users.

TABLE 2-1 File System Manager Permission Levels

Administrative Privilege Level	Description
<code>com.sun.netstorage.fsmgr.config</code>	User has unlimited access.
<code>com.sun.netstorage.fsmgr.operator.media</code>	User can add or remove libraries, add or remove stand-alone drives, reserve volume serial names (VSNs), import VSNs, load and unload VSNs, export VSNs, and so on.
<code>com.sun.netstorage.fsmgr.operator.sam.control</code>	User can start, stop, or idle archiving operations.
<code>com.sun.netstorage.fsmgr.operator.file</code>	User can start or stop staging, and can restore a file system.
<code>com.sun.netstorage.fsmgr.operator.filesystem</code>	User can mount or unmount a file system, edit mount options, and perform file system checks (<code>fsck</code>).

To specify full or partial configuration privileges for a user, add the following line to the `/etc/user_attr` file:

```
account-name:::auths=privilege-level
```

account-name is the name of the user's account and *privilege-level* is the level of authorization that you want to assign to the user.

For example, to assign full privileges (privilege level `com.sun.netstorage.fsmgr.config`) for user account `bobsmith`, add the following line to the `/etc/user_attr` file:

```
bobsmith:::auths=com.sun.netstorage.fsmgr.config
```

To assign `bobsmith` privileges only for staging and restoring file systems (privilege level `com.sun.netstorage.fsmgr.operator.file`) and exporting, importing, and assigning VSNs (privilege level `com.sun.netstorage.operator.media`), add the following line to the `/etc/user_attr` file:

```
bobsmith:::auths=com.sun.netstorage.fsmgr.operator.file,  
com.sun.netstorage.fsmgr.operator.media
```

Creating an Account for Multiple Users

You can create a generic File System Manager account that can be used by multiple users, and then add a role with privileges that only some of those users can access.

1. Use the `useradd` and `passwd` commands to add the account.

For example, to add a user account called `guest` for multiple users, type the following:

```
# /usr/sbin/useradd guest  
# /usr/bin/passwd guest
```

2. Use the `roleadd` and `passwd` commands to add the role.

To create a role called `admin` with special privileges within the `guest` account, type the following:

```
# /usr/sbin/roleadd admin  
# /usr/bin/passwd admin
```

3. Specify the privilege levels in the `/etc/user_attr` file.

To assign the `admin` role privileges to restore and stage file systems, add the following lines to the `/etc/user_attr` file:

```
admin:::auths=com.sun.netstorage.fsmgr.operator.file  
guest:::type=normal;roles=admin
```

In this example, when a user logs in as `guest`, File System Manager prompts the user to select either No Role or Admin. If users know the Admin role password, they can select Admin, enter the Admin password, and have privileges to restore and stage file systems. All other users must select No Role and have read-only privileges.

Because multiple users with the same privilege level can be logged in to the software concurrently, there is a risk of one user's changes overwriting another user's previous changes. To prevent this, develop policies about who can make changes and how to notify others.

▼ To Add an Additional Server for File System Manager Access

File System Manager is set up by default to manage the server on which it is installed. It can also be used to manage other servers running Sun StorageTek QFS software, but those additional servers must first be configured to allow File System Manager access.

1. **Outside of the browser interface, use the `telnet` utility to connect to the server you want to add. Log in as `root`.**
2. **Use the `fsmadm(1M) add` command to add the management station (the system on which the File System Manager software is installed) to the list of hosts that can remotely administer this server.**

Only hosts that are added to the list through this command can remotely administer the server.

For example:

```
# fsmadm add management_station.sample.com
```

3. **To ensure that the management station is successfully added, use the `fsmadm(1M) list` command and verify that your management station is listed in the output.**
4. **Log in to the File System Manager browser interface as an administrator user.**
5. **From the Servers page, click Add.**
The Add Server window is displayed.
6. **In the Server Name or IP Address field, type the name or the IP address of the new server.**
7. **Click OK.**

Setting the Session Timeout

The Java Web Console framework has a default session timeout of 15 minutes. The File System Manager installation program changes the session timeout to 60 minutes. You can change the session timeout to a different value, but it is recommended that you not set it to a value greater than 60 minutes in order to preserve security.

To change the session timeout value, enter the following command on the management station:

```
/opt/SUNWfsmgr/bin/fsmgr session <timeout-in-minutes>
```

For example, to change the timeout value to 45 minutes, type:

```
/opt/SUNWfsmgr/bin/fsmgr session 45
```

Function of the mcf File

The master configuration file (mcf), located in `/etc/opt/SUNWsamfs/mcf`, describes all devices that are under the control of, or used by, the Sun StorageTek 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`.

An mcf file consists of lines of specification code divided into six columns, or fields, as shown in [CODE EXAMPLE 2-1](#).

CODE EXAMPLE 2-1 Fields in an mcf File

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

Follow these rules when entering data in the mcf file:

- Enter either space or tab characters between the fields in the file.
- You can include comment lines in an mcf file. Comment lines start with a pound character (#).
- Some fields are optional. Use a dash character (-) to indicate that an optional field contains no meaningful information.

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

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

- “The Equipment Identifier Field” on page 35
- “The Equipment Ordinal Field” on page 36
- “The Equipment Type Field” on page 36
- “The Family Set Field” on page 37
- “The Device State Field” on page 38
- “The Additional Parameters Field” on page 38

The Equipment Identifier Field

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

- The file system name. If this field contains a file system name, it must be identical to the 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. File system names must start with an alphabetic character and can contain only alphabetic characters, numeric characters, or underscore (`_`) characters.
- The `nodev` keyword. The keyword `nodev` indicates that the system on which the `mcf` file resides is being used as a client host in a Sun StorageTek QFS shared file system on a Solaris host. Do not use this keyword if your file system is in a Sun Cluster environment. This keyword can appear in this field only as the Equipment Identifier for one or more metadata devices that reside on the metadata server. For more information about creating an `mcf` file for the members of a Sun StorageTek QFS shared file system, see the *Sun StorageTek QFS Installation and Upgrade Guide*.
- A disk partition or slice description. A `/dev/` entry in this field identifies a disk partition or slice.
- An automated library or optical drive description. A `/dev/samst` entry identifies an automated library or optical drive. If you are configuring a network attached automated library, see the *Sun StorageTek Storage Archive Manager Installation and Upgrade Guide* for more information.

- A tape drive description. This entry can be in one of two forms:
 - 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.

If the Equipment Identifier 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.

The Equipment Ordinal Field

For each row in the `mcf` file, the Equipment Ordinal (*eq*) field must contain a numeric identifier for the file system component or device being defined. Specify a unique integer between 1 and 65534, inclusive. This is a required field.

The Equipment Type Field

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

The Equipment Type field in a Sun StorageTek QFS or SAM-QFS file system can contain any of the values shown in [TABLE 2-2](#).

TABLE 2-2 Sun StorageTek QFS or SAM-QFS Equipment Type Field

Equipment Type Field Content	Meaning
ma	Defines a Sun StorageTek QFS file system that stores metadata on a separate device (an <code>mm</code> device).
ms	Defines a Sun StorageTek QFS file system that stores both data and metadata on the same device.
md	Defines a striped or round-robin device for storing file data.
mm	Defines a metadata device for storing inode and other nondata information. You can specify multiple metadata devices. Metadata (including inodes, directories, allocation maps, and so on) on Sun StorageTek QFS <code>ma</code> 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.

TABLE 2-2 Sun StorageTek QFS or SAM-QFS Equipment Type Field (*Continued*)

Equipment Type Field Content	Meaning
<code>gXXX</code>	<p>Defines a striped group data device. Striped groups start with the letter <code>g</code> followed by a number. The number must be an integer between 0 and 127, inclusive; for example, <code>g12</code>.</p> <p>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. <code>md</code>, <code>mr</code>, and <code>gXXX</code> devices cannot be mixed in one file system.</p> <p>Data can be striped (if all groups contain the same number of devices) or round-robin between groups. The default is round-robin.</p>

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 `mc f(4)` man page.

The Family Set Field

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

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

The lines that define the disk devices in a file system must all contain the same family set name. The software uses the family set 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 `sammkfs(1M)` command is issued. You can change this name by using the `-F` and `-R` options together in the `samfsck(1M)` command. For more information about the `sammkfs(1M)` command, see the `sammkfs(1M)` man page. For more information about the `samfsck(1M)` command, see the `samfsck(1M)` man page.

The lines that define the devices in an automated library, as well as the lines that define the devices in the library's associated drives, must contain the same family set name.

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

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 File System Manager software, any related comments will also be deleted from the `mcf` file.

The Device State Field

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

The Additional Parameters Field

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

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

Examples of `mcf` Files

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

CODE EXAMPLE 2-2 shows an `mcf` file for a Sun StorageTek QFS file system with two striped groups.

CODE EXAMPLE 2-2 Example `mcf` File Showing Striped Groups

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

CODE EXAMPLE 2-3 shows an `mcf` file with three Sun StorageTek QFS file systems.

CODE EXAMPLE 2-3 Example `mcf` File Showing Three SAM-QFS 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  -
#
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  -
```

CODE EXAMPLE 2-4 shows an `mcf` file with one SAM-QFS file system that uses `md` devices. This `mcf` file also defines a tape library.

CODE EXAMPLE 2-4 Example `mcf` File Showing a File System and a 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
```

For more examples of file system configurations, see the *Sun StorageTek QFS Installation and Upgrade Guide*.

Interactions Among File Settings, Options, and Directives

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

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

For information about the various ways to specify mount options, see [“Setting Up Mount Parameters” on page 60](#).

Initializing a File System

If you want to create a new file system or replace an old or damaged file system, you must initialize it using the `sammkfs(1M)` command.

The `sammkfs(1M)` command constructs new file systems, and its `-a allocation-unit` option enables you to specify the DAU setting.

Version 4U2 and later releases of the Sun StorageTek QFS software support two different superblock designs. In [CODE EXAMPLE 2-5](#), the `samfsinfo(1M)` command output shows that the `samfs1` file system is using a version 2 superblock.

CODE EXAMPLE 2-5 `samfsinfo(1M)` Command Example

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

Be aware of the following operational and feature differences that pertain to these superblocks:

- Releases before 4U0 support only the version 1 superblock design.
- The 4U0 and later releases support the version 2 superblock. If you installed the 4U0 software as an upgrade, you must use the 4U0 or later `sammkfs(1M)` command to reinitialize your existing file systems before you attempt to use any of the features that depend on the version 2 superblock. Certain features, such as access control lists (ACLs) and the Sun StorageTek QFS shared file system, are supported only in the version 2 superblock. Reinitializing a file system is described as a step in the software installation upgrade process, but you can do this any time after the software is installed.



Caution – File systems that use a version 2 superblock cannot revert to a release before 4U0. You cannot use 4U6 release software to create a version 1 superblock.

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

The following example shows the `sammkfs(1M)` command in its simplest form, with the file system name as its only argument:

```
# sammkfs samqfs1
```

The preceding command builds a version 2 superblock for a stand-alone Sun StorageTek QFS or SAM-QFS file system.

For more information about the `sammkfs(1M)` command, its options, and the implications of the version 1 and version 2 superblocks, see the `sammkfs(1M)` man page. For information about using the `sammkfs(1M)` command to initialize a shared Sun StorageTek QFS file system, see the *Sun StorageTek QFS Installation and Upgrade Guide*.

Configuration Examples

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

- [“To Create a Sun StorageTek QFS Round-Robin Disk Configuration” on page 42](#)
- [“To Create a Sun StorageTek QFS Striped Disk Configuration” on page 44](#)
- [“To Create a Sun StorageTek QFS Striped Group Configuration” on page 45](#)

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

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

▼ To Create a Sun StorageTek QFS Round-Robin Disk Configuration

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

This procedure assumes the following:

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

1. Use an editor to create the `mcf` file, as shown in [CODE EXAMPLE 2-6](#).

CODE EXAMPLE 2-6 Example Sun StorageTek QFS Round-Robin `mcf` File

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

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

```
# mkdir /qfs
```

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

The following example uses the default 64-kilobyte DAU:

```
# sammkfs qfs1
```

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

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

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

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

```
# mount /qfs
```

▼ To Create a Sun StorageTek QFS 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 ordinal 11 only.
- The data devices consist of four disks attached to four controllers. Each disk is on a separate controller.

1. Use an editor to create the `mcf` file, as shown in [CODE EXAMPLE 2-7](#).

CODE EXAMPLE 2-7 Example Sun StorageTek QFS Striped Disk `mcf` File

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

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

```
# mkdir /qfs
```

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

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

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

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

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

The Sun StorageTek 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. Use the `mount(1M)` command to mount the file system:

```
# mount /qfs
```

▼ To Create a Sun StorageTek QFS 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 bitmaps. If the striped group has n devices, n multiplied by the DAU is the minimum allocation. Only one bit in the bitmaps is used to represent $n \times$ DAU.

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

minimum-disk-space-allocated = DAU \times number-of-disks-in-the-group



Caution – 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 \times 4 disks = 512 kilobytes in length) still use 512 kilobytes of disk space. Files larger than 512 kilobytes have space allocated for them as needed in total space increments of 512 kilobytes.

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

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

This procedure assumes the following:

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

1. Use an editor to create the `mcf` file, as shown in [CODE EXAMPLE 2-8](#).

CODE EXAMPLE 2-8 Example Sun StorageTek QFS Striped Group `mcf` File

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

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

```
# mkdir /qfs
```

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

The following example sets the DAU size to 128 kilobytes:

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

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

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

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

This `/etc/vfstab` file sets the stripe width using the `stripe=` option. In this example, there are two striped groups, `g0` and `g1`. With the `stripe=0` specification, 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(1M)` command.

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

```
# mount /qfs
```


Performing Operations

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

- [“Viewing Files and File Attributes” on page 49](#)
 - [“Propagating Configuration File Changes to the System” on page 54](#)
 - [“Setting Up Mount Parameters” on page 60](#)
 - [“Unmounting a File System” on page 63](#)
 - [“Adding Disk Cache to a File System” on page 64](#)
 - [“Re-creating a File System” on page 66](#)
-

Viewing Files and File Attributes

The attributes specific to Sun StorageTek QFS file systems include both user settings and general file states. This section describes these characteristics and describes how to view them using the `sls` command.

File Attributes and File States

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

A user can specify the following commands to set attributes:

- `archive(1)`
- `ssum(1)`
- `release(1)`

- `segment(1)`
- `setfa(1)`
- `stage(1)`

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

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

[TABLE 3-1](#) shows the user-specified attributes that are listed in the inode.

TABLE 3-1 User-Specified File Attributes

File Attribute	Description
<code>setfa -D</code>	Marks the file for direct I/O.
<code>setfa -gn</code>	Marks the file for allocation on striped group <i>n</i> .
<code>setfa -sm</code>	Marks the file for allocation with a stripe width of <i>m</i> .

Note – There are a number of additional user-specified file attributes that are specific to Sun StorageTek SAM archiving functionality. See the *Sun StorageTek Storage Archive Manager File System Configuration and Administration Guide* for more information.

You can set the attributes shown in [TABLE 3-1](#) on both files and directories. After directory attributes are set, files that are created in the directory inherit all the directory attributes. Files created before an attribute is applied to the parent directory do not inherit directory attributes.

If you have the WORM-FS package installed, you can also apply WORM (write once read many) attributes to a file, and set the file’s retention period. See [“Configuring WORM-FS File Systems” on page 192](#) for details.

Displaying File Information

The Sun StorageTek QFS `s1s(1)` command extends the standard UNIX `ls(1)` command and provides more information about a file. [CODE EXAMPLE 3-1](#) shows detailed `s1s(1)` command output that displays the inode information for file `hgc2`.

CODE EXAMPLE 3-1 `s1s(1)` Output in a SAM-QFS Environment

```
# s1s -D hgc2
hgc2:
mode: -rw-r--r--  links: 1  owner: root      group: other
length: 14971  admin id: 0  inode: 30.5
archdone;
segments 3, offline 0, archdone 3, damaged 0;
copy 1: ---- Jun 13 17:14      2239a.48  lt MFJ192
copy 2: ---- Jun 13 17:15      9e37.48  lt AA0006
access: Jun 13 17:08  modification: Jun 13 17:08
changed: Jun 13 17:08  attributes: Jun 13 17:10
creation: Jun 13 17:08  residence: Jun 13 17:08
```

About the `s1s(1)` Output

[TABLE 3-2](#) describes the meaning of each row of `s1s(1)` output shown in [CODE EXAMPLE 3-1](#).

Note – Lines that pertain to archiving only appear in `s1s(1)` output only in a Sun StorageTek SAM environment.

TABLE 3-2 `s1s(1)` Output Explanation

Line Number	Tag	Content
1	mode:	The file's mode and permissions, the number of hard links to the file, the owner of the file, and the group to which the owner belongs.

TABLE 3-2 `sls(1)` Output Explanation (*Continued*)

Line Number	Tag	Content
2	<code>length:</code>	<p>The file's length in bytes, the file's admin ID number, and the file's inode number.</p> <p>By default, the admin ID number is 0. If this number is greater than 0, it indicates the file's accounting category for counting files and blocks. You can set this number to a value greater than 0 even when file system quotas are not enabled on this file system. For information about file system quotas, see "Administering File System Quotas" on page 97.</p> <p>The inode number is a two-part number that contains the inode number itself, followed by a period (.), followed by the inode generation number.</p>
3	<code>archdone;</code>	<p>The file attributes specific to the file. For more information about this line, see the <code>sls(1)</code> man page.</p>
4	<code>segments</code>	<p>The segment index information. This line does not appear unless the file is a segment index. The general format for this line is as follows:</p> <p><code>segments n, offline o, archdone a, damaged d;</code></p> <ul style="list-style-type: none">• <code>segments n</code> shows the total number of data segments for this file. In this example, there are 3.• <code>offline o</code> shows the number of data segments offline. In this example, there are no offline segments.• <code>archdone a</code> shows the number of segments for which the archiving requirements have been met. In this example, there are 3.• <code>damaged d</code> shows the number of damaged segments. In this example, there are no damaged segments.

TABLE 3-2 `sls(1)` Output Explanation (*Continued*)

Line Number	Tag	Content
5, 6	<code>copy 1:</code> <code>copy 2:</code>	<p>Archive copy lines. The <code>sls(1)</code> command displays one archive copy line for each active or expired archive copy.</p> <p>The four positions in this line indicate the following:</p> <ul style="list-style-type: none">1 – Either an expired or an active entry.<ul style="list-style-type: none">• An <code>S</code> indicates that the archive copy is expired. That is, the file was modified and this archive copy is a previous version of the file.• A <code>U</code> indicates that the copy has been unarchived. Unarchiving is the process by which archive entries for files or directories are deleted.• A dash (<code>-</code>) indicates that the archive copy is active and valid.2 – Whether the archive copy is to be rearchived.<ul style="list-style-type: none">• An <code>r</code> indicates that the archive copy is scheduled to be rearchived by the archiver.• A dash (<code>-</code>) indicates that the archive copy is not to be rearchived by the archiver.3 – Unused.4 – Whether the copy is damaged or undamaged.<ul style="list-style-type: none">• A <code>D</code> indicates that the archive copy is damaged. A damaged archive copy is not a candidate for staging.• A dash (<code>-</code>) indicates that the archive copy is not damaged. It is a candidate for staging. <p>The format of the rest of the archive copy line is as follows:</p> <ul style="list-style-type: none">• The date and time the archive copy was written to the archive media.• Two hexadecimal numbers separated by a decimal point (<code>.</code>). The first hexadecimal number (<code>2239a</code>) indicates the position of the beginning of the archive file on the cartridge. The second hexadecimal number (<code>48</code>) is the file byte offset (divided by 512) of this copy in the archive file.• The media type and the volume serial name (VSN) where the archive copy resides.
7	<code>access:</code>	The time the file was last accessed and modified.
8	<code>changed:</code>	The time the file content and the file's attributes were last changed.
9	<code>creation:</code>	The time the file was created and became resident in the file system.

About the Retention Line

If you are using the optional WORM-FS package, a retention line will also appear in the `sls(1)` output. The format of the retention line is as follows:

```
retention: active retention-period: 3y 0d 0h 0m
```

This indicates whether a retention period has been set for this file and, if so, what its length is. The `retention-end` date indicates the date on which the retention period expires. For more information about using the WORM-FS feature, see [“Configuring WORM-FS File Systems” on page 192](#).

Propagating Configuration File Changes to the System

This section describes how to propagate configuration file changes throughout the system. The procedures describe the propagation of changes for the following files:

- `mcf(4)`
- `defaults.conf`
- `archiver.cmd` (SAM-QFS file systems only)
- `stager.cmd` (SAM-QFS file systems only)
- shared hosts file (Sun StorageTek QFS shared and SAM-QFS shared file systems only)

You must perform these procedures under the following circumstances:

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

The following sections describe these procedures:

- [“To Change `mcf` or `defaults.conf` Information in a Sun StorageTek QFS Sun Cluster Environment” on page 55](#)
- [“To Change `mcf` or `defaults.conf` File System Information in a SAM-QFS Environment” on page 55](#)
- [“To Change `mcf` or `defaults.conf` Removable Media Drive Information” on page 56](#)
- [“Changing the Shared Hosts File” on page 58](#)

- [“To Add New Entries or Change Existing Entries” on page 58](#)
- [“To Change Host Names, Reorder Entries, or Insert Entries” on page 59](#)

▼ To Change `mcf` or `defaults.conf` Information in a Sun StorageTek QFS Sun Cluster Environment

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

1. Use `vi(1)` or another editor to edit the file and change the file system information.
2. If you are changing the `mcf` file, use the `sam-fsd(1M)` command to check the `mcf` file for errors:

```
# sam-fsd
```

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

3. Use the `samd(1M) config` command to propagate the `mcf(4)` or `defaults.conf` file changes:

```
# samd config
```

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

▼ To Change `mcf` or `defaults.conf` File System Information in a SAM-QFS Environment

1. Use `vi(1)` or another editor to edit the file and change the file system information.
2. If you are changing the `mcf` file, use the `sam-fsd(1M)` command to check the `mcf` file for errors:

```
# sam-fsd
```

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

3. **If you are removing or changing information related to one or more file systems, issue a `samcmd(1M) aridle` command to idle the archiver for each affected file system defined in the `mcf` file.**

Use this command in the following format:

```
samcmd aridle fs.fsname
```

For *fsname*, specify the name of the file system.

4. **If you are removing or changing information related to one or more drives, issue a `samcmd(1M) idle` command to idle the archiver for each equipment ordinal assigned to each affected drive in the `mcf` file.**

Use this command in the following format:

```
samcmd idle eq
```

For *eq*, specify the Equipment Ordinal number of the drive.

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

For instructions on unmounting the file system, see [“Unmounting a File System” on page 63](#).

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

```
# samd config
```

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

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

▼ To Change `mcf` or `defaults.conf` Removable Media Drive Information

1. **Edit the file and change the removable media drive information.**

2. If you are changing the `mcf` file, use the `sam-fsd(1M)` command to check the `mcf` file for errors:

```
# sam-fsd
```

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

3. If you are removing or changing information related to one or more file systems, issue a `samcmd(1M)` `aridle` command to idle the archiver for each affected file system defined in the `mcf` file.

Use this command in the following format:

```
samcmd aridle fs.fsname
```

For *fsname*, specify the name of the file system.

4. If you are removing or changing information related to one or more drives, issue a `samcmd(1M)` `idle` command for the Equipment Ordinal number assigned to each affected drive in the `mcf` file.

Use this command in the following format:

```
samcmd idle eq
```

For *eq*, specify the Equipment Ordinal number of the drive.

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

```
# samd stop
```

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

```
# samd config
```

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

```
# samd start
```

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

Changing the Shared Hosts File

You can add new host entries or change existing entries in the shared hosts file without unmounting the file system. To change host names, reorder the entries, or insert an entry, you must first unmount the file system. Use the following procedures to change the shared hosts file.

▼ To Add New Entries or Change Existing Entries

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

1. **If you do not know the host that is acting as the metadata server, issue the `samsharefs(1M)` *family-set-name* command to display the name of the metadata server.**

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

For example:

```
# samsharefs sharefs1
```

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

For example:

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

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

For example:

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

4. **Use `vi(1)` or another editor to edit the temporary working file.**

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

5. **Save and close the temporary working file.**

6. Copy the temporary working file to the `SUNWsamfs` directory.

For example:

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

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

For example:

```
# samsharefs -u sharefs1
```

▼ To Change Host Names, Reorder Entries, or Insert Entries

The procedure for changing host names, reordering entries, or inserting entries in the shared hosts file requires that you unmount the file system.

1. If you do not know the host that is acting as the metadata server, issue the `samsharefs(1M) -R family-set-name` command to display the name of the metadata server.

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

For example:

```
# samsharefs -R sharefs1
```

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

For example:

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

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

For example:

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

5. Use `vi(1)` or another editor to edit the temporary working file.
6. Save and close the shared hosts file.

7. Copy the new shared hosts file to the `SUNWsamfs` directory.

For example:

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

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

For example:

```
# samsharefs -uR sharefs1
```

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

Setting Up Mount Parameters

You can mount a Sun StorageTek QFS file system by using the Solaris OS `mount(1M)` command.

Mount parameters are used to manipulate file system characteristics. There are several ways to specify mount parameters. Methods at the top of the hierarchy override methods lower in the hierarchy. You can specify mount options in the following ways, listed in hierarchical order from the top down:

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

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

The following subsections describe ways to specify mount options. For information about specific mount options, see [Appendix B “Mount Options in a Sun StorageTek QFS Shared File System” on page 251](#). The *Sun StorageTek QFS Installation and Upgrade Guide* also includes information about mounting a file system.

The mount(1M) Command

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

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

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

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

If you are mounting a Sun StorageTek QFS or SAM-QFS shared file system, you must mount the file system on the metadata server first, and then mount the file system on each participating client host. Include the `shared` option with the `mount` command, and remember that the command must be identical on the metadata server and on the participating hosts.

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

The /etc/vfstab File

Each Sun StorageTek QFS file system that is defined in the `mcf` file must have a line in the `/etc/vfstab` Solaris OS system file. This is required for mounting the file system.

The following is an example of a file system line in the `/etc/vfstab` file:

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

From left to right, the fields shown indicate the following:

- The file system family set name.
- The file system to `samfsck(1M)`.

- The mount point.
- The file system type. This is always `samfs`, even for Sun StorageTek QFS file systems.
- The `samfsck(1M)` pass.
- Mount-at-boot options.
- Mount parameters, separated by commas, without intervening spaces.

The fields in the `/etc/vfstab` file must be separated by either space or tab characters.

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

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

The `samfs.cmd` File

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

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

The mount parameters that can be specified in the `samfs.cmd` file are nearly identical to those that you can specify in the `/etc/vfstab` file or as arguments to the `-o` option with the `mount(1M)` command. The possible mount parameters you can specify pertain to I/O settings, read-ahead, write-behind, the stripe width, various storage and archive management settings, WORM-FS, Qwrite, and other features. For more information about the mount parameters that can be specified in this file, see the `samfs.cmd(4)` man page.

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

For a directive that applies globally to all file systems, place the line before any `fs =` line. For a directive that is specific to a particular file system, start the line with `fs =` and place it after all global directives. Directives specific to a particular file system override global directives.

[CODE EXAMPLE 3-2](#) shows a sample `samfs.cmd` file that sets the low-water and high-water marks for disk cache utilization for all file systems and specifies individualized parameters for two specific file systems.

CODE EXAMPLE 3-2 Example `samfs.cmd` File

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

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

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

Unmounting a File System

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

▼ To Unmount a Stand-alone QFS or SAM-QFS File System

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

```
# umount /samqfs
```

Before unmounting a file system, you may want to `unshare(1M)` it if it has been shared out for NFS use. After unsharing, try issuing the `umount` command a second time. If this fails and it is necessary to unmount the file system, use the `-f` option to the `umount(1M)` command. The `-f` option forces a file system to unmount.

For more information on unmounting procedures, see the *Sun StorageTek QFS Installation and Upgrade Guide*.

▼ To Unmount a Sun StorageTek QFS Shared File System

Follow the instructions in [“To Unmount a Shared File System”](#) on page 70.

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(1M)` 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 StorageTek 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>s on eq <eq>
does not match <id> in mcf
```

▼ To Add Disk Cache to a File System

1. **Use the `umount(1M)` command to unmount the file system you want to expand.**
If the file system is shared, unmount the file system on all client hosts and then on the metadata server. You can then perform the remaining steps in this procedure on the metadata server.
For more information about unmounting a file system, see [“Unmounting a File System” on page 63](#).
2. **If you want to rename the file system during this procedure, use the `samfsck(1M)` command with its `-R` and `-F` options to rename the file system.**
For more information about this command, see the `samfsck(1M)` man page.
3. **Edit the `/etc/opt/SUNWsamfs/mcf` file to add the disk cache.**
4. **Issue the `sam-fsd(1M)` command to 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. **Issue the `samd(1M) config` command to propagate the `mcf` file changes to the system:**

```
# samd config
```

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

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

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

```
# samgrowfs samfs1
```

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

7. Mount the file system.

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

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

Re-creating a File System

In order to do any of the following, you must re-create the file system:

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

This section describes this procedure.

▼ 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 `mcf`, `archiver.cmd`, `defaults.conf`, `samfs.cmd`, or `inquiry.conf`. Back up these files for all file systems in your Sun StorageTek 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, there is no need to back them up again now. If, however, you need to back up your file

systems to preserve information created since the last dump file was created, do so now. For information about how to create a dump file using `qfscdump`, see the *Sun StorageTek QFS Installation and Upgrade Guide*.

3. Unmount the file system.

For instructions, see [“Unmounting a File System” on page 63](#).

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

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

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

For more information, see [“Adding Disk Cache to a File System” on page 64](#).

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

```
# sam-fsd
```

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

7. Issue the `samd(1M) config` command to propagate the `mcf` file changes to the system:

```
# samd config
```

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

8. Issue the `sammkfs(1M)` command to re-create the file system.

For example, the following command creates `samfs10`:

```
# sammkfs samfs10
```

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

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

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

11. Use the `qfsrestore(1M)` command, or use File System Manager, to restore each file.

Restore from the dump file you had or from the dump file created in Step 1.

For more information, see the `qfsdump(1M)` man page or the File System Manager online help.

Configuring a Sun StorageTek QFS Shared File System

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

- [“Mounting and Unmounting Sun StorageTek QFS Shared File Systems” on page 69](#)
- [“Converting an Unshared File System to a Shared File System” on page 71](#)
- [“Converting a Shared File System to an Unshared File System” on page 75](#)
- [“Adding or Removing a Client Host” on page 77](#)
- [“Changing the Metadata Server in a Sun StorageTek QFS Environment” on page 90](#)
- [“Client-Server Communications in a Sun StorageTek QFS Shared File System” on page 94](#)

Mounting and Unmounting Sun StorageTek QFS Shared File Systems

When you mount or unmount a Sun StorageTek QFS 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 containing mount options and copy it to all of the hosts.

For more information about mounting Sun StorageTek QFS shared file systems, see [“Mount Options in a Sun StorageTek QFS Shared File System” on page 251](#) and see the `mount_samfs(1M)` man page. For more information about mounting and unmounting file systems, see [Chapter 3, “Performing Operations” on page 49](#).

▼ To Mount a Shared File System

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

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

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

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

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

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

▼ To Unmount a Shared File System

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

For example:

```
client# umount /samqfs
```

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

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

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 may require you to issue the `umount(1M)` command a second time. If the file system still does not unmount, use `unshare(1M)`, `fuser(1M)`, or another command in conjunction with the `umount(1M)` command. For more information on unmounting procedures, see the `umount(1M)` man page and the *Sun StorageTek QFS Installation and Upgrade Guide*.

You can also use the `-o await_clients #` flag with the `umount` command. This causes 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.

Converting an Unshared File System to a Shared File System

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

To convert an unshared Sun StorageTek QFS file system to a Sun StorageTek QFS shared file system, you must perform the conversion first on the metadata server and then on each client. This section describes these procedures.

▼ To Perform a Conversion on the 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. Back up all site-customized system files and configuration files.

Depending on your software, these files might include `mcf`, `archiver.cmd`, `defaults.conf`, `samfs.cmd`, or `inquiry.conf`. 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. 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, there is no need to back them up again now.

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

For instructions, see [“Unmounting a File System”](#) on page 63.

5. Use the `samfsck(1M) -S -F family-set-name` command to convert the file system to a Sun StorageTek QFS shared file system.

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

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

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

For example:

CODE EXAMPLE 4-1 mcf File for Shared File System, sharefs1

```
# Equipment           Eq Eq   Family   Dev   Add
# Identifier          Ord Type Set      State Params
# -----
sharefs1              10  ma   sharefs1 on    shared
/dev/dsk/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. Edit the `/etc/vfstab` file to add the `shared` keyword in the file system's **Mount Parameters** field.

For example:

CODE EXAMPLE 4-2 /etc/vfstab File Example

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

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

For example:

CODE EXAMPLE 4-3 Sun StorageTek QFS Shared File System Hosts File Example

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host Host IPServerNotServer
# Name AddressesPriorityUsedHost
# -----
titan titan-ge0 1-server
tethys tethys-ge02-server
```

See the *Sun StorageTek QFS Installation and Upgrade Guide* for more information about creating the hosts configuration file.

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

For example:

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

Note – This command might issue an error message, which can be ignored.

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

```
# samd config
```

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

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

▼ To Perform a Conversion on Each Client

1. **Use the `mkdir(1)` command to 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 StorageTek QFS shared host systems have multiple host interfaces. The local hosts configuration file defines the host interfaces that the metadata server and the client hosts can use when accessing the file system. You use this file to specify how file system traffic should flow over public and private networks in your environment.

[CODE EXAMPLE 4-4](#) shows a sample local hosts configuration file.

CODE EXAMPLE 4-4 File `hosts.sharefs1.local`

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

For more information on creating the local hosts file, see [“Creating the Local Hosts Configuration File” on page 87](#).

3. If you want to move files from an existing Sun StorageTek QFS file system into a new Sun StorageTek 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, there is no need to back them up again now.

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

For instructions, see [“Unmounting a File System” on page 63](#).

5. Edit the `/etc/vfstab` file to 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. [CODE EXAMPLE 4-5](#) shows a sample.

CODE EXAMPLE 4-5 Sun StorageTek QFS Shared File System Hosts File Example

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host Host IPServerNotServer
# Name AddressesPriorityUsedHost
# -----
titan titan-ge0 1- server
tethys tethys-ge0 2- server
```

For more information about creating the hosts configuration file, see the *Sun StorageTek QFS Installation and Upgrade Guide*.

Converting a Shared File System to an Unshared File System

To convert a Sun StorageTek QFS shared file system to an unshared Sun StorageTek QFS file system, you must perform the conversion first on each client and then on the metadata server. This section describes these procedures.

▼ To Perform a Conversion on Each Client

1. Use the `umount(1M)` command to unmount the file system.
For instructions, see [“Unmounting a File System” on page 63](#).
2. Delete the file system’s entry from the `/etc/opt/SUNWsamfs/mcf` file.
3. Delete the file system’s entry from the `/etc/vfstab` file.
4. Run the `samd(1M) config` command:

```
# samd config
```

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

5. Delete the mount point for the file system.

▼ To Perform a Conversion on the Server

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

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 `mcf(4)`, `archiver.cmd`, `defaults.conf`, `samfs.cmd`, `inquiry.conf`, and so on. Back up these files for all file systems. Also make sure that you have backup copies of files in the `/etc/opt/SUNWsamfs` directory and files in the `/var/opt/SUNWsamfs` directory.
3. **If you want to move files from an existing Sun StorageTek QFS shared file system into a new Sun StorageTek 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, there is no need to back them up again now.

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

For instructions, see [“Unmounting a File System” on page 63](#).
5. **Run the `samfsck(1M) -F -U file-system-name` to convert the Sun StorageTek QFS shared file system to an unshared file system.**

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

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

6. **Edit the `/etc/opt/SUNWsamfs/mcf` file to 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. Edit the `/etc/vfstab` file to 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. Run the `samd(1M) config` command:

```
# samd config
```

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

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

Adding or Removing a Client Host

The following subsections provide instructions for adding and removing client host systems in a Sun StorageTek QFS shared file system:

- [“To Add a Client Host” on page 77](#)
- [“To Remove a Client Host” on page 82](#)
- [“Updating the `mcf` file in a Sun StorageTek QFS Shared Environment” on page 84](#)
- [“Creating the Local Hosts Configuration File” on page 87](#)

▼ To Add a Client Host

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

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

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

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

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

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

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

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

[CODE EXAMPLE 4-6](#) shows this step.

CODE EXAMPLE 4-6 `hosts.sharefs1` Before Editing

```
# vi /etc/opt/SUNWsamfs/hosts.sharefs1
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host Host IPServerNotServer
# Name AddressesPriorityUsedHost
# -----
titan 172.16.0.129 1-server
tethys 172.16.0.130 2-
mimas mimas --
dione dione --
```

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

CODE EXAMPLE 4-7 shows the file after addition of the line for helene as the last line.

CODE EXAMPLE 4-7 hosts.sharefs1 After Editing

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host Host IPServerNotServer
# Name AddressesPriorityUsedHost
# -----
titan 172.16.0.129 1-server
tethys 172.16.0.130 2-
mimas mimas - -
dione dione - -
helene helene - -
```

5. Use the `samsharefs(1M)` 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 StorageTek QFS shared file system is mounted, as follows:

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

```
# samsharefs -u sharefs1
```

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

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

The client host `helene` is now recognized.

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

7. Use the `format(1M)` command to verify the presence of client host disks.

8. Update the `mcf` file on the client host.

Before a host system can access or mount a shared file system, it must have that file system defined in its `mcf` file. The `mcf` file must be updated to match all client hosts in the Sun StorageTek QFS shared file system. The file system and disk declaration information must have the same data for the Family Set Name, Equipment Ordinal,

and Equipment Type as the configuration on the metadata server. The `mcf` files on the client hosts must also include the shared keyword. The device names, however, can change, since controller assignments will probably differ from host to host.

For information on how to edit the `mcf` file, see [“Updating the `mcf` file in a Sun StorageTek QFS Shared Environment” on page 84](#).

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

```
# samd config
```

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

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

You might want to perform this step if your Sun StorageTek QFS shared host systems have multiple host interfaces. The local hosts configuration file defines the host interfaces that the metadata server and the client hosts can use when accessing the file 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 87](#).

11. Issue the `samd(1M) config` command on the client host:

```
# samd config
```

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

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

To accomplish this, use the `ps(1)` and `grep(1)` commands as shown in [CODE EXAMPLE 4-8](#).

CODE EXAMPLE 4-8 Output from the `ps(1)` Command

```
# ps -ef | grep sam-sharefsd  
root 26167 26158 0 18:35:20 ?          0:00 sam-sharefsd sharefs1  
root 27808 27018 0 10:48:46 pts/21    0:00 grep sam-sharefsd
```

[CODE EXAMPLE 4-8](#) shows that the `sam-sharefsd` daemon is active for the `sharefs1` file system. If the output returned on your system does not show that the `sam-sharefsd` daemon is active for your Sun StorageTek QFS shared file system, perform the diagnostic procedures described in [“Troubleshooting a Failed or Hung `sammkfs\(1M\)` or `mount\(1M\)` Command in a Shared File System” on page 231](#).

13. If the new Sun StorageTek QFS shared file system does not already have a mount point, use the `mkdir(1)` command to make the directory for the mount point.

For example:

```
# mkdir /sharefs1
```

14. Issue the `chmod(1M)` command to give the mount point the 755 set of permissions.

For example:

```
# chmod 755 /sharefs1
```

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

15. Modify the `/etc/vfstab` file.

You must have an entry in the `/etc/vfstab` file for the Sun StorageTek 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 StorageTek QFS shared file system to automatically mount at boot, do the following:
 - Type `yes` in the `Mt@boot` field.
 - Add the `bg` mount option in the `Mt params` field. The `bg` mount option mounts the file system in the background if the metadata server is not responding.

[CODE EXAMPLE 4-9](#) shows the `shared` and `bg` entries in the `Mt params` field.

CODE EXAMPLE 4-9 `/etc/vfstab` File Example

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

16. Issue the `df(1M)` command on the metadata server to verify that the file system is mounted on the metadata server.

For example:

```
# df -k
```

The file system should be included in the displayed list.

17. From the client host, issue the `mount(1M)` command to mount the Sun StorageTek QFS shared file system.

For example:

```
# mount /sharefs1
```

For more information about mounting Sun StorageTek QFS shared file systems, see [“Mount Options in a Sun StorageTek QFS Shared File System” on page 251](#), or see the `mount_samfs(1M)` man page.

▼ To Remove a Client Host

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

Note – You can use the `samsharefs(1M)` command to verify that you are, indeed, logged in to the metadata server or a client host.

2. Use the `umount(1M)` command to unmount the Sun StorageTek QFS shared file system on each client host on which the Sun StorageTek QFS shared file system is mounted.

For example:

```
client# umount sharefs1
```

3. Use the `umount(1M)` command to unmount the Sun StorageTek QFS shared file system on the metadata server.

For example:

```
metaserver# umount sharefs1
```

4. If you have not already done so, log in as superuser to the metadata server for the Sun StorageTek QFS shared file system.
5. Use the `samsharefs(1M)` command to obtain the current configuration information.

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

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

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

[CODE EXAMPLE 4-10](#) shows the file before the client host is deleted.

CODE EXAMPLE 4-10 `hosts.sharefs1` Before Deleting a Client Host

```
# vi /etc/opt/SUNWsamfs/hosts.sharefs1
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host Host IPServerNotServer
# Name AddressesPriorityUsedHost
# -----
titan 172.16.0.129 1-server
tethys 172.16.0.130 2-
mimas mimas - -
dione dione - -
helene helene --
```

7. Use the editor to delete the client host or hosts that are no longer to be supported.

[CODE EXAMPLE 4-11](#) shows the file after the line for `helene` has been deleted.

CODE EXAMPLE 4-11 `hosts.sharefs1` After Deleting a Client Host

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host Host IPServerNotServer
# Name AddressesPriorityUsedHost
# -----
titan 172.16.0.129 1-server
tethys 172.16.0.130 2-
mimas mimas - -
dione dione - -
```

8. Use the `samsharefs(1M) -R -u` command to update the current hosts information.

For example:

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

The host `helene` has been removed.

9. Use the `samsharefs(1M) -R` command to display the current configuration.

For example:

```
# samsharefs -R sharefs1
```

10. Use the `mount(1M)` command to mount the Sun StorageTek QFS shared file system, first on the metadata server and then on each client host in the file system.

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

Updating the `mcf` file in a Sun StorageTek QFS Shared Environment

The `samfsconfig(1M)` command generates configuration information that can help you to identify the devices included in the Sun StorageTek QFS shared file system. You can then use this information to update the `mcf` files on each client host.

Enter a separate `samfsconfig(1M)` command on each client host. Note that the controller number might not be the same controller number as on the metadata server because the controller numbers are assigned by each client host.

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

Example 1. [CODE EXAMPLE 4-12](#) shows how the `samfsconfig(1M)` command is used to retrieve device information for family set `sharefs1` on client `tethys`. Because `tethys` is a potential metadata server, it is connected to the same metadata disks as `titan`, another metadata server in the shared file system.

CODE EXAMPLE 4-12 `samfsconfig(1M)` Command Example on `tethys`

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

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

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

[CODE EXAMPLE 4-13](#) shows the resulting `mcf` file.

CODE EXAMPLE 4-13 `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/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
```

Example 2. [CODE EXAMPLE 4-14](#) shows how the `samfsconfig(1M)` command is 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.

CODE EXAMPLE 4-14 `samfsconfig(1M)` Command Example on `mimas`

```
mimas# samfsconfig /dev/dsk/*
#
# Family Set 'sharefs1' Created Wed Jun 27 19:33:50 2001
#
# Missing slices
# Ordinal 0
# /dev/dsk/c1t50020F2300005D22d0s6 12 mr sharefs1 on
# /dev/dsk/c1t50020F2300006099d0s6 13 mr sharefs1 on
# /dev/dsk/c1t50020F230000651Cd0s6 14 mr sharefs1 on
```

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

- Create a file system Family Set declaration line, beginning with `sharefs1`, in the `mcf` file for client host `mimas`. Enter the `shared` keyword in the Additional Parameters field of the file system Family Set declaration line.
- Create one or more `nodev` lines for each missing Equipment Ordinal 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.

[CODE EXAMPLE 4-15](#) shows the resulting `mcf` file for `mimas`.

CODE EXAMPLE 4-15 `mcf` File for Client Host `mimas`

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

Creating the Local Hosts Configuration File

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

```
/etc/opt/SUNWsamfs/hosts.family-set-name.local
```

Comments are permitted in the local hosts configuration file. Comment lines must begin with a pound character (#). Characters to the right of the pound character are ignored.

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

TABLE 4-1 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 StorageTek QFS shared file system.
Host Interfaces	<p>This field must contain a comma-separated list of host interface addresses. This field can be created from the output received from the <code>ifconfig(1M)</code> - a 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 Sun StorageTek QFS shared file system, each client host obtains the list of metadata server IP addresses from the metadata server host.

The metadata server and the client hosts use both the `/etc/opt/SUNWsamfs/hosts.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 (note that *client*, as in *network client*, is used to refer to both client hosts and the metadata server host):

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

To examine this file, issue the `samsharefs(1M)` command from the metadata server or from a potential metadata server.

2. The client searches its files for a `hosts.fsname.local` file.
3. 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:
 - a. It compares the list of addresses for the metadata server from both the `/etc/opt/SUNWsamfs/hosts.fsname` file on the metadata server and the `hosts.fsname.local` file.
 - b. It builds a list of addresses that are present in both places, and then it attempts to connect to each of these addresses, in turn, until it succeeds. If the order of the addresses differs in these files, the client uses the ordering in the `hosts.fsname.local` file.

Example. [CODE EXAMPLE 4-16](#) shows an example hosts file listing four hosts.

CODE EXAMPLE 4-16 Sun StorageTek QFS Shared File System Hosts File Example

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host Host IPServerNotServer
# Name AddressesPriorityUsedHost
# -----
titan 172.16.0.129 1-server
tethys 172.16.0.130 2-
mimas mimas - -
dione dione - -
```

[FIGURE 4-1](#) shows the interfaces to these systems.

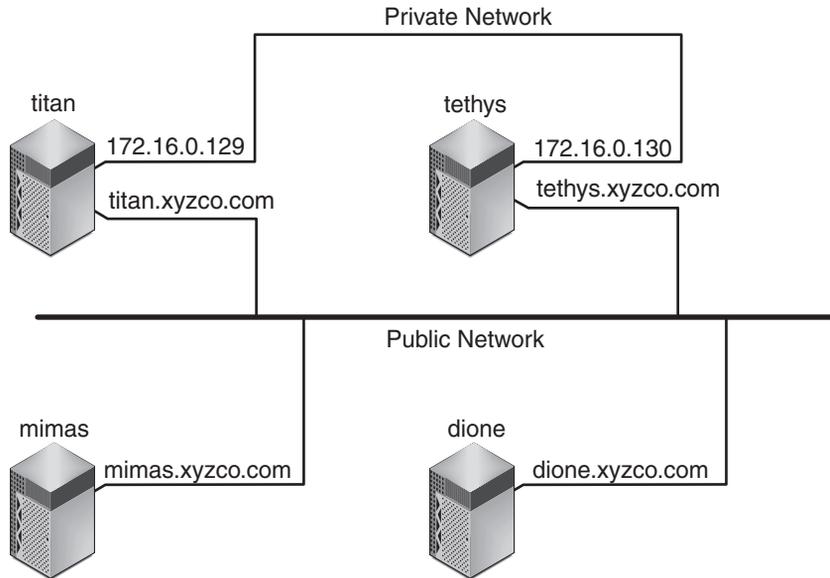


FIGURE 4-1 Network Interfaces

Systems titan and tethys share a private network connection with interfaces 172.16.0.129 and 172.16.0.130. To guarantee that titan and tethys always communicate over their private network connection, the system administrator has created identical copies of `/etc/opt/SUNWsamfs/hosts.sharefs1.local` on each system. [CODE EXAMPLE 4-17](#) shows the information in these files.

CODE EXAMPLE 4-17 File `hosts.sharefs1.local` 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
```

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. [CODE EXAMPLE 4-18](#) shows the information in these files.

CODE EXAMPLE 4-18 File `hosts.sharefs1.local` 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 in a Sun StorageTek QFS Environment

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

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.

Choose one of the following procedures depending on whether the existing metadata server is available at the time the change is being performed:

- [“To Change the Metadata Server When the Metadata Server Is Available” on page 91](#)
- [“To Change the Metadata Server When the Metadata Server Is Unavailable” on page 91](#)

▼ To Change the Metadata Server When the Metadata Server Is Available

- On the existing metadata server, issue the `samsharefs(1M) -s` command to declare the new metadata server.

For example:

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

Note – In Sun SAM-QFS environments, all archiving operations should be stopped on the metadata server before issuing this command.

▼ To Change the Metadata Server When the Metadata Server Is Unavailable

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(1)`, 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.

Use the following steps to change the metadata server:

- 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).

[CODE EXAMPLE 4-19](#) shows the key sequence to use from the `kadb` prompt.

CODE EXAMPLE 4-19 Key Sequence for Ensuring that the Metadata Server Cannot Restart from the `kadb` Prompt

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

[CODE EXAMPLE 4-20](#) shows the key sequence to use from the PROM prompt.

CODE EXAMPLE 4-20 Key Sequence for Ensuring that the Metadata Server Cannot Restart from the PROM Prompt

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

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

2. **From the new (potential) metadata server, wait for at least the period of the maximum lease time, and then issue the `samsharefs(1M)` command.**

For example:

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

The wait ensures that all client leases expire before you issue the `samsharefs(1M)` command. If you are uncertain as to whether the lease time has expired, bring up the `samu(1M) N` display. For information about `samu(1M)`, see [“Using the `samu\(1M\)` Operator Utility” on page 259](#). For information about leases and their durations, see [“Using Leases in a Sun StorageTek QFS Shared File System: the `rdlease=n`, `wrllease=n`, and `aplease=n` Options” on page 254](#).



Caution – If you use the `-R` option to the `samsharefs(1M)` command on a mounted file system to change the metadata server host, you must first stop, disable, and disconnect the active metadata server. Failure to do so can cause file system corruption.

3. **(Optional) Unmount the file system.**

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

Use the procedure in [“To Unmount a Sun StorageTek QFS Shared File System” on page 64](#).

4. (Optional) Issue the `samfsck(1M)` command to perform a file system check.

If the metadata server of a Sun StorageTek QFS shared file system crashes, the server should be rebooted and the file system should be unmounted on all clients before `samfsck(1M)` is run. The server and clients preallocate blocks before changing the length of files. The `samfsck(1M)` command cleans up files that have extra blocks allocated, and these extra blocks might contain data. If such a cleaned-up file is awaiting a size update from the client, the file will be missing those blocks when the client continues. As a result, the file will be missing data, and the missed data will read as zeroes.

Changing the Metadata Server in a SAM-QFS Environment

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

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.

▼ To Change the Metadata Server in a SAM-QFS Environment

Sun StorageTek SAM can only be running on one host at any time. This procedure assumes that both systems are up at the time of the transfer. In this example we are moving Sun StorageTek SAM 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 Sun StorageTek SAM archiving processes on host A by carrying out the following steps:

a. Run `samcmd aridle` and `samcmd stidle` 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.

This can be done with `samcmd eq idle`, where *eq* is the equipment ordinal of the drive. This 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, Sun StorageTek SAM has been halted and file system failover to host B can be performed.

2. Start Sun StorageTek SAM on host B by running `samd config` on host B.

This 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 StorageTek QFS shared client applications waiting for stages must reissue the stage requests.

Host B should now be fully functioning as the Sun StorageTek SAM server and metadata server for all Sun StorageTek QFS file systems.

Client-Server Communications in a Sun StorageTek QFS Shared File System

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

The system generates the following messages to describe status conditions:

```
SAM-FS: Shared server is not responding.
```

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

```
SAM-FS: Shared server is responding.
```

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

```
SAM-FS: Shared server is not mounted.
```

When the Sun StorageTek QFS 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 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 *Solaris Tunable Parameters Reference Manual*. The parameter that controls the size of the directory name lookup cache is `ncsize`.

Administering File System Quotas

This chapter describes how to enable and administer file system quotas. This chapter contains the following sections:

- [“Overview” on page 97](#)
 - [“Enabling Quotas” on page 100](#)
 - [“Checking Quotas” on page 109](#)
 - [“Changing and Removing Quotas” on page 111](#)
-

Overview

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.

The following subsections provide background information about using quotas:

- [“Types of Quotas, Quota Files, and Quota Records” on page 98](#)
- [“Soft Limits and Hard Limits” on page 99](#)
- [“Disk Blocks and File Limits” on page 99](#)

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 ["To Enable or Change Limits for Users, Groups, or Admin Sets Using an Existing Quota File"](#) on page 107.

Record one is the record in the quota file for user one, group one, or admin set ID one, depending on the type of quota file. You can edit record one and all subsequent records in order to set different quotas for different users. [TABLE 5-1](#) shows the quota file names and the quotas they enable in `/root`.

TABLE 5-1 Quota File Names

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

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

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

$$\begin{aligned} (\text{highest-ID} + 1) \times 128 &= x \\ x / 4096 &= \text{zero quota file size} \end{aligned}$$

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 is aborted. In this case, 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.

You, the administrator, can use the `samquota(1M)` command to see the timer value. The `squota(1)` command is a user version of the `samquota(1M)` command. The `squota(1)` user command contains options that users can specify to obtain information about quotas that pertain to them.

Disk Blocks and File Limits

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

File system quotas are expressed in terms of the number of 512-byte blocks that a user can allocate. However, disk space is allocated to user files in terms of DAUs. The DAU setting is specified by the `-a allocation-unit` option to the `sammkfs(1M)` command. It is preferable to set a block quota to a multiple of the file system DAU. If this is not done, users can allocate only up to the block count, rounded down to the nearest DAU. See [“Enabling Default Quota Values” on page 106](#) 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.

The following subsections provide more details on how to configure a file system to use quotas and how to enable quotas.

Guidelines for Setting Up Quotas

Before you enable quotas, you should determine how much disk space and how many inodes to allocate to each user. If you want to be sure that the total file system space is never exceeded, you can divide the total size of the file system 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(1)` command is for end users. It enables them to retrieve quota information for themselves on a user, group, or admin set basis.
- The `samquota(1M)` command is for system administrators. It enables you to retrieve quota information or to set quotas. Use the `-U`, `-G`, and `-A` options for a user, a group, or an admin set, respectively. [CODE EXAMPLE 5-1](#) shows this.

CODE EXAMPLE 5-1 Using `samquota(1M)` to Retrieve Information

```
# samquota -U janet /mount-point    #Prints a user quota
# samquota -G pubs /mount-point     #Prints a group quota
# samquota -A 99 /mount-point       #Prints an admin set quota
```

▼ To Configure a New File System to Use Quotas

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 [“To Configure an Existing File System to Use Quotas”](#) on page 102.

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 StorageTek QFS Installation and Upgrade Guide*, or use the examples in “[Configuration Examples](#)” on page 42 to create the `mc.f` file, create the mount point, initialize the file system, and so on.

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

For example:

```
# mount /qfs1
```

4. Use the `dd(1M)` 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, use the following command:

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

- To create group quotas, use the following command:

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

- To create user quotas, use the following command:

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

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

5. Use the `umount(1M)` command to unmount the file system 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. For more information about the `umount(1M)` command, see the `umount(1M)` man page.

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

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

```
# samfsck -F qfs1
```

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

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(1M)` command, and quotas are enabled automatically when the system detects the presence of quota files.

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

8. Use the `samquota(1M)` command to set quotas for users, groups, or admin sets.

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

▼ To Configure an Existing File System to Use Quotas

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 [“To Configure a New File System to Use Quotas” on page 100](#).

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. Use the `su(1)` command to become superuser.
2. Use the `mount(1M)` 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. Use the `cd(1)` command to change to the root directory of the file system for which quotas are to be enabled.

For example:

```
# cd /oldfs1
```

4. Use the `ls(1) -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`, `.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(1M)` command to create the quota files.

Determine the highest existing ID numbers of the types of quotas you wish to enforce. Make the initial, zero, quota files large enough to hold the 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:

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

$$131200/4096 = 32.031\dots$$

Use the following command:

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

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

6. Use the `umount(1M)` command to unmount the file system in which the quota files have been created.

For example:

```
# umount /oldfs1
```

The file system 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 a File System” on page 63](#).

7. Use the `samfsck(1M) -F` command to perform a file system check.

This command updates records allocated in the quota files with correct, current usage information.

For example:

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

8. Use the `mount(1M)` command to remount the file system in which the quota files have been created.

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

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(1M)` 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(1M) -F` command to update the quota file usage counts before again remounting the file system with quotas enabled.

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

9. Use the `samquota(1M)` command to set quotas for users, groups, or admin sets.

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

▼ To Assign Admin Set IDs to Directories and Files

1. Use the `su(1)` command to become superuser.

2. Set the admin IDs.

Use the `samchaid(1M)` command to change the admin set IDs for the directory or file, as follows:

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

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

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

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

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

Setting Infinite Quotas

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

▼ To Set an Infinite Quota

- Use the `samquota(1M)` command to set the quota limit to zero.

For example:

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

You can use the `samquota(1M)` command to set infinite quotas for particular users, groups, or admin set IDs by setting zero values for all hard and soft limits.

[CODE EXAMPLE 5-2](#) shows how to set infinite quotas.

CODE EXAMPLE 5-2 Setting Infinite Quotas

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

			Online Limits			Total Limits		
	Type	ID	In Use	Soft	Hard	In Use	Soft	Hard
/sam6								
Files	group	101	339	0	0	339	0	0
Blocks	group	101	248	0	0	2614	0	0
Grace period				0s			0s	
---> Infinite quotas in effect.								

Enabling Default Quota Values

You can use the `samquota(1M)` command to enable a default quota for a user, group, or admin set. This is accomplished through default limits in user, group, or admin set zero.

▼ To Enable Default Quota Values for Users, Groups, or Admin Sets

● Use the `samquota(1M)` command.

For example, the following `samquota(1M)` command sets default quotas for all admin set IDs:

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

The preceding 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 occurs, for example, if the admin group already has blocks assigned to it.

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

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

Enabling Limits

You can use the `samquota(1M)` command to enable a set of limits for a particular user, group, or admin set.

▼ To Enable Limits for Users, Groups, or Admin Sets

- Use the `samquota(1M)` command.

[CODE EXAMPLE 5-3](#) shows commands that enable limits for users, groups, and admin sets, respectively.

CODE EXAMPLE 5-3 Quota Commands

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

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

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

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

1. Use the `samquota(1M)` command to retrieve a quota file, and direct the output to a temporary file.

Use the `-e` option with one or more of the following additional options: `-U userID`, `-G groupID`, or `-A adminsetID`.

[CODE EXAMPLE 5-4](#) shows how to create and retrieve file `quota.group` to use as a template.

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

CODE EXAMPLE 5-4 File `quota.group`

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

CODE EXAMPLE 5-4 File quota.group (Continued)

```
# Online Limits Total Limits
# soft hard soft hard
# Files
# Blocks
# Grace Periods
#
samquota -G 101 \
-f 200:s:o -f 300:h:o -f 200:s:t -f 300:h:t \
-b 40000:s:o -b 60000:h:o -b 40000000:s:t -b 60000000:h:t \
-t 0s:o -t 0s:t /sam6
```

2. Use an editor to edit the temporary file you just created.

CODE EXAMPLE 5-5 shows the file that was generated in [Step 1](#) opened in the vi(1) editor. Group ID 101 has been changed to 102. This has the effect of generating a command to copy the quotas set for group 101 to group 102.

CODE EXAMPLE 5-5 File quota.group After Editing

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

3. Save the file and exit the editor.

4. To apply the changes made in the editor, execute the file using the shell.

For example:

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

In this example, the -x option directs the shell to echo the commands it executes. You can omit the -x option if desired.

Checking Quotas

After you have enabled disk and inode quotas, you can check these quotas. The `samquota(1M)` command is an administrator command that generates a quota report on an individual user, group, or admin set. The `squota(1)` command is a user command that enables users to check their own individual quotas.

▼ To Check for Exceeded Quotas

1. **Become superuser.**
2. **Use the `samquota(1M)` command to display the quotas in effect for mounted file systems.**
 - To display user quotas, specify the following command:

```
# samquota -U userID [ file ]
```

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

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

Example 1. [CODE EXAMPLE 5-6](#) retrieves user `hm1259`'s quota statistics in the `sam6` file system on the server and displays output indicating that this user is not exceeding the quota.

CODE EXAMPLE 5-6 Checking for Exceeded Quotas for User `hm1259`

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

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

Example 2. [CODE EXAMPLE 5-7](#) retrieves user `memil`'s quota statistics in all mounted Sun StorageTek QFS file systems and displays output indicating that this user is exceeding the quota. Note the plus sign (+) in the `Blocks` row of the output. The plus sign would appear in the `Files` row, too, if the soft quota limit were being exceeded for files.

CODE EXAMPLE 5-7 Checking for Exceeded Quotas for User `memil`

```
# samquota -U memil

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

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

- To display group quotas, specify the following command:

```
# samquota -G groupID [ file ]
```

For `groupID`, specify the numeric group ID or the group name for the group of users whose quotas are being examined. For `file`, specify a specific file system for the selected 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
```

- To display admin set quotas, specify the following command:

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

For *adminsetID*, specify the numeric admin set ID of the site-specific administrator set whose quotas are being examined. For *file*, specify a specific file system for the selected 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.

For example, the following command retrieves user quota statistics for the admin set 457 in all mounted Sun StorageTek QFS 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. The following subsections describe how to change and remove quotas:

- [“To Change the Grace Period” on page 111](#)
- [“Changing the Grace Period Expiration” on page 113](#)
- [“To Inhibit Additional File System Resource Allocations” on page 115](#)
- [“To Remove a File System’s Quotas” on page 117](#)
- [“To Correct Quotas” on page 118](#)

▼ To Change the Grace Period

You can use the `samquota(1M)` command to change the soft time limit grace period.

1. Use the `samquota(1M)` command to retrieve quota statistics for a user, group, or admin set.

See [“To Check for Exceeded Quotas” on page 109](#) for instructions.

Example. [CODE EXAMPLE 5-8](#) retrieves information about group `sam` and shows that this group is over its soft limit.

CODE EXAMPLE 5-8 Exceeding a 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

CODE EXAMPLE 5-8 Exceeding a Soft Limit (Continued)

```
Blocks group 101 41888* 40000 60000000 43208 60000000 60000000
Grace period 1w
---> Online soft limits under enforcement (since 30s ago)
```

2. Examine the output from the `samquota(1M)` command and determine what the new limits should be.
3. Use the `samquota(1M)` command to change the soft time limit grace period.

[CODE EXAMPLE 5-9](#) shows using the `samquota(1M)` command options to use.

CODE EXAMPLE 5-9 Using `samquota(1M)` to Change Soft Time Limit Grace Periods

```
# samquota -U userID -t interval file
# samquota -G groupID -t interval file
# samquota -A adminID -t interval file
```

The arguments for these commands are as follows:

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

For example, suppose that you wanted to change the grace period for user `mem11`. [CODE EXAMPLE 5-10](#) shows the `samquota(1M)` command that you would use to verify the quotas and its output.

CODE EXAMPLE 5-10 Changing the Grace Period

```
# samquota -U mem1 /sam6

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

You would enter the following command to shorten the grace period.

```
# samquota -U mem1 -t 1d /sam6
```

CODE EXAMPLE 5-11 shows the `samquota(1M)` command that you would then use to verify the new quotas.

CODE EXAMPLE 5-11 Verifying the New Quotas

```
# samquota -U mem1 /sam6

          Type      ID      In Use      Online Limits      Total Limits
          Type      ID      In Use      Soft      Hard      In Use      Soft      Hard
/sam6
Files   user 130967          4          500      750          4          500      750
Blocks user 130967      41016+    40000    50000     41016     50000    50000
Grace period                1d                          0s
---> Warning:  online soft limits to be enforced in 23h58m31s
```

Changing the Grace Period Expiration

If a user has exceeded 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(1M)` 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.

[CODE EXAMPLE 5-12](#) 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`.

CODE EXAMPLE 5-12 Clearing the Timer

```
# samquota -G sam -x clear /sam6
Setting Grace Timer: continue? y
# samquota -G sam /sam6
```

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

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

- **Reset the grace period timer** – When an expiration period is reset, the timer is reset, and the grace period restarts.

[CODE EXAMPLE 5-13](#) resets the grace period.

CODE EXAMPLE 5-13 Resetting the Grace Period Timer

```
# samquota -G sam -x reset /sam6
Setting Grace Timer: continue? y
# samquota -G sam /sam6
```

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

```
---> Warning: online soft limits to be enforced in 6d23h59m52s
```

- **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.

[CODE EXAMPLE 5-14](#) sets a very long expiration period.

CODE EXAMPLE 5-14 Setting a Very Long Grace Period

```
# samquota -G sam -x 52w /sam6
Setting Grace Timer: continue? y
# samquota -G sam /sam6
```

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

CODE EXAMPLE 5-14 Setting a Very Long Grace Period (Continued)

```
/sam6
Files group 101 32 2000 2000 32 2000 2000
Blocks group 101 41888+ 40000 60000000 43208 60000000 60000000
Grace period 1w 1w
--> Warning: online soft limits to be enforced in 51w6d23h59m54s
```

- **Expire the grace period timer** – The timer is set to expire immediately.

[CODE EXAMPLE 5-15](#) expires the grace period.

CODE EXAMPLE 5-15 Expiring the Grace Period Timer

```
# samquota -G sam -x expire /sam6
Setting Grace Timer: continue? y
# samquota -G sam /sam6

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

▼ To Inhibit Additional File System Resource Allocations

When the file system detects that quota values are not consistent for a user, group, or admin set, it prevents that user, group, or admin set from using any more system resources. You can inhibit file system resource allocations by creating inconsistent quota values. 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, and from there they can become the default values for new users, groups, or admin set IDs.

The following procedure shows how to inhibit further system resource allocations for a user, group, or admin set.

1. **Become superuser.**

2. Obtain, save, and examine current quota information.

[CODE EXAMPLE 5-16](#) shows how to use the `samquota(1M)` command to retrieve current group quota information for group `sam` and write it to a backup file.

CODE EXAMPLE 5-16 Retrieving Group Quota Information

```
# 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 userID` option in place of the `-G` option. To obtain quota information about an admin set quota, specify the `-A adminID` option in place of the `-G` option.

3. Use the `samquota(1M)` command to set soft quotas to nonzero quotas and hard quotas to zero quotas.

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. Use the `samquota(1M)` command to verify your changes.

[CODE EXAMPLE 5-17](#) shows this.

CODE EXAMPLE 5-17 Verifying Changed Quotas

```
# samquota -G sam /sam6

Type ID In Use Online Limits In Use Total Limits
Soft Hard Soft Hard
/sam6
Files group 101 32! 1 0 32! 1 0
```

CODE EXAMPLE 5-17 Verifying Changed Quotas (Continued)

```
Blocks group 101 41888! 1 0 43208! 1 0
Grace period 1w 1w
--> Quota values inconsistent; zero quotas in effect.
```

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

5. Use the `sh(1)` and `samquota(1M)` commands to restore the group's quota to what it was before the file/block allocation was inhibited and then to verify the changed quotas.

[CODE EXAMPLE 5-18](#) shows these commands.

CODE EXAMPLE 5-18 Restoring the Group Quota

```
# sh restore.quota.sam
Setting Grace Timer: continue? y
Setting Grace Timer: continue? y
# samquota -G sam /sam6

          Type      ID      In Use      Online Limits      Total Limits
          Soft      Hard      In Use      Soft      Hard
/sam6
Files group 101      32      2000      2000      32      2000      2000
Blocks group 101      41888+ 40000 60000000 43208 60000000 60000000
Grace period 1w      1w
--> Warning: online soft limits to be enforced in 6d23h59m54s
```

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

▼ To Remove a File System's Quotas

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

1. Use the `su(1)` command to become superuser.
2. (Optional) Use a text editor to add the `noquota` mount option to the `/etc/vfstab` or `samfs.cmd` file.

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

3. If the file system is mounted, use the `umount(1M)` command to unmount the file system.

For example:

```
# umount /myfs
```

If you have difficulty unmounting the file system, see [“Unmounting a File System” on page 63](#).

4. Remount the file system using the `mount(1M)` command.

If you did not perform [Step 2](#), include the `noquota` option with the `mount(1M)` command.

For example:

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

5. Dispense with 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(1M)` command with its `-F` option on the file system, and remount the file system again with the `noquota` mount option removed.
- If you do not expect to reinstate the quota feature at a later date, or if you want to reclaim the space consumed by the quota files, use the `rm(1)` command to remove the `.quota_u`, `.quota_g`, and `.quota_a` files. For example:

```
# rm /myfs/.quota_[agu]
```

▼ To Correct Quotas

1. Become superuser.
2. If the file system is mounted, use the `umount(1M)` command to unmount the file system.

For example:

```
# umount /myfs
```

If you have difficulty unmounting the file system, see [“Unmounting a File System” on page 63](#).

3. Use the `samfsck(1M) -F` command to perform a file system check.

The `samfsck(1M)` command updates records allocated in the quota files with correct, current usage information.

For example:

```
# samfsck -F myfs
```

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

For example:

```
# mount /myfs
```


Configuring Sun StorageTek QFS in a Sun Cluster Environment

This chapter describes how the Sun StorageTek QFS software works in a Sun Cluster environment. It also provides configuration examples for a Sun StorageTek QFS shared file system in a Sun Cluster environment and for an unshared Sun StorageTek QFS file system in a Sun Cluster environment.

This chapter contains the following sections:

- [“Before You Begin” on page 122](#)
- [“Restrictions” on page 123](#)
- [“How the Sun Cluster System and the Sun StorageTek QFS Software Interact” on page 124](#)
- [“Sun StorageTek QFS Support for Solaris Volume Manager for Sun Cluster” on page 125](#)
- [“About Configuration Examples” on page 132](#)
- [“Configuring a Sun StorageTek QFS Shared File System in a Sun Cluster Environment” on page 134](#)
- [“Configuring an Unshared File System in a Sun Cluster Environment” on page 144](#)
- [“Configuring Shared Clients Outside the Cluster” on page 165](#)
- [“Changing the Sun StorageTek QFS Configuration” on page 177](#)
- [“High-Availability Sun StorageTek SAM Configuration Using Sun Cluster” on page 183](#)

Before You Begin

With versions 4U2 and later of the Sun StorageTek QFS software, you can install a Sun StorageTek QFS file system in a Sun Cluster environment and configure the file system for high availability. The configuration method you use varies, depending on whether your file system is shared or unshared.

This chapter assumes that you are an experienced user of both the Sun StorageTek QFS software and the Sun Cluster environment. It also assumes you have performed either or both of the following:

- Configured file systems as highly available scalable, or configured failover resources under Sun Cluster software control
- Installed and configured both Sun StorageTek QFS stand-alone and Sun StorageTek QFS shared file systems

It is recommended that you read the following documentation before continuing with this chapter:

- The following chapters of the *Sun StorageTek QFS Installation and Upgrade Guide*:
 - Chapter 2, which provides hardware and software prerequisite information for the Sun StorageTek QFS software in a Sun Cluster environment
 - Chapter 4, which describes additional tasks for installing a Sun StorageTek QFS file system on a Sun Cluster system for high availability
- The following Sun Cluster software documentation:
 - *Sun Cluster Concepts Guide for Solaris OS*, which introduces the Sun Cluster environment. In particular, review the following sections:
 - Local Disks
 - Global Devices
 - Device ID (DID)
 - Disk Device Groups
 - Disk Device Group Failover
 - Local and Global Namespaces
 - Cluster File Systems
 - HAStoragePlus Resource Type
 - Volume Managers
 - The *Sun Cluster Software Installation Guide for Solaris OS*, which describes the steps for installing the Sun Cluster software

- The *Sun Cluster Data Services Planning and Administration Guide for Solaris OS*, which explains how to plan your Sun Cluster environment to include various data services
- The *Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS*, which describes how to use the Sun StorageTek QFS shared file system with the Sun Cluster data service for Oracle Real Application Clusters (RACs)

Note – The File System Manager software can also be used to control file systems in Sun Cluster environments. It recognizes and identifies cluster nodes and automatically prompts you to add other cluster nodes when adding a server. You have the option of creating non-archiving highly available (HA) shared or stand-alone Sun StorageTek QFS file systems on nodes within a Sun Cluster configuration. See the File System Manager online Help for more information.

Restrictions

The following restrictions apply to the Sun StorageTek QFS software in a Sun Cluster environment:

- The following restrictions apply to shared file systems:
 - They can be configured only under the `SUNW.qfs` resource type. They cannot be configured under the `HASStoragePlus` resource type.
 - They can be configured as scalable file systems used only by the Sun Cluster data service for Oracle Real Application Clusters.
- The following restrictions apply to unshared file systems:
 - They can be configured only under the `HASStoragePlus` resource type. They cannot be configured under the `SUNW.qfs` resource type.
 - Any failover application that is supported with Sun Cluster software is supported with unshared Sun StorageTek QFS software as a highly available local file system. Support is not limited to specific applications.
 - An unshared Sun StorageTek QFS file system cannot be the underlying native file system in a Sun Cluster system.

Note – Failback is not supported as a feature of the `SUNW.qfs` agent.

Note – Although installing a Sun StorageTek QFS file system in a Sun Cluster environment improves reliability and decreases or eliminates unplanned downtime, it does not eliminate planned downtime. In order to maintain the health of the file system, the Sun StorageTek QFS software may need to be brought down occasionally to run the `samfsck` process. It also needs to be shut down in order to apply software patches or updates.

How the Sun Cluster System and the Sun StorageTek QFS Software Interact

The shared file system uses Sun Cluster disk identifier (DID) support to enable data access by the Sun Cluster data service for Oracle Real Application Clusters. The unshared file system uses global device volume support and volume manager-controlled volume support to enable data access by failover applications supported by the Sun Cluster system.

Data Access With a Shared File System

With DID support, each device that is under the control of the Sun Cluster system, whether it is multipathed or not, is assigned a unique DID. For every unique DID device, there is a corresponding global device. The Sun StorageTek QFS shared file system can be configured on redundant storage that consists only of DID devices (`/dev/did/*`), where DID devices are accessible only on nodes that have a direct connection to the device through a host bus adapter (HBA).

Configuring the Sun StorageTek QFS shared file system on DID devices and configuring the `SUNW.qfs` resource type for use with the file system makes the file system's shared metadata server highly available. The Sun Cluster data service for Oracle Real Application Clusters can then access data from within the file system. Additionally, the Sun StorageTek QFS Sun Cluster agent can then automatically relocate the metadata server for the file system as necessary.

Note – Beginning with version 4U6 of the Sun StorageTek QFS software you can also have shared clients outside of the cluster in a Sun Cluster environment. For complete configuration instructions, see [“Configuring Shared Clients Outside the Cluster” on page 165](#).

Data Access With an Unshared File System

A global device is the Sun Cluster system's mechanism for accessing an underlying DID device from any node within the Sun Cluster system, assuming that the nodes hosting the DID device are available. Global devices and volume manager-controlled volumes can be made accessible from every node in the Sun Cluster system. The unshared Sun StorageTek QFS file system can be configured on redundant storage that consists of either raw global devices (`/dev/global/*`) or volume manager-controlled volumes.

Configuring the unshared file system on these global devices or volume manager-controlled devices and configuring the `HASStoragePlus` resource type for use with the file system makes the file system highly available with the ability to fail over to other nodes.

Sun StorageTek QFS Support for Solaris Volume Manager for Sun Cluster

In the 4U4 release of Sun StorageTek QFS, support was added for Solaris Volume Manager for Sun Cluster, which is an extension to Solaris™ Volume Manager that is bundled with the Solaris 9 and Solaris 10 OS releases. Sun StorageTek QFS only supports Solaris Volume Manager for Sun Cluster on Solaris 10.

Sun StorageTek QFS support for Solaris Volume Manager for Sun Cluster was introduced to take advantage of shared Sun StorageTek QFS host-based mirroring as well as Oracle's implementation for application binary recovery (ABR) and direct mirror reads (DMR) for Oracle RAC-based applications.

Use of Solaris Volume Manager for Sun Cluster with Sun StorageTek QFS requires Sun Cluster software and an additional unbundled software package included with the Sun Cluster software.

With this addition of Solaris Volume Manager for Sun Cluster support, four new mount options were introduced. These mount options are only available if Sun StorageTek QFS detects that it is configured on Solaris Volume Manager for Sun Cluster. The mount options are:

- `abr` - Application binary recovery enabled
- `dmr` - Direct mirror reads enabled
- `noabr` - Application binary recovery disabled
- `nodmr` - Direct mirror reads disabled

The following is a configuration example for using Sun StorageTek QFS with Solaris Volume Manager for Sun Cluster.

In this example below, it is assumed the following configuration has been done:

- Sun Cluster has been installed, and has been taken out of install mode.
- The additional packages have been installed to enable the Sun Cluster Oracle RAC Framework.
- The additional Sun Cluster `SUNWscmd` package has been installed to enable Solaris Volume Manager for Sun Cluster.
- The Oracle RAC Framework Resource Group has been created, and is online.

In this example there are three shared Sun StorageTek QFS file systems:

- **CRS** - This file system is required for the Oracle 10G release configuration of Oracle RAC cluster-ready services (CRS). See *Sun Cluster Concepts* for additional information.
- **Data** - This file system is used to contain Oracle-related files: Oracle system files, Oracle control files, and Oracle data files.
- **Redo** - This file system is used to contain Oracle Redo, Oracle Archive, and Flash Back files and logs.

▼ To Configure a File System With Solaris Volume Manager for Sun Cluster

1. Create the `metadb` on each node.

For example:

```
# metadb -a -f -c3 /dev/rdisk/c0t0d0s7
```

2. Create the disk group on one node.

For example:

```
# metaset -s datadg -M -a -h scNode-A scNode-B
```

3. Run `scdidadm` to obtain devices on one node.

For example:

```
scNode-A # scdidadm -l
13 scNode-A: /dev/rdisk/c6t600C0FF00000000000332B62CF3A6B00d0 /dev/did/rdisk/d13
14 scNode-A: /dev/rdisk/c6t600C0FF00000000000876E950F1FD9600d0 /dev/did/rdisk/d14
15 scNode-A: /dev/rdisk/c6t600C0FF00000000000876E9124FAF9C00d0 /dev/did/rdisk/d15
16 scNode-A: /dev/rdisk/c6t600C0FF00000000000332B28488B5700d0 /dev/did/rdisk/d16
17 scNode-A: /dev/rdisk/c6t600C0FF0000000000086DB474EC5DE900d0 /dev/did/rdisk/d17
18 scNode-A: /dev/rdisk/c6t600C0FF00000000000876E975EDA6A000d0 /dev/did/rdisk/d18
19 scNode-A: /dev/rdisk/c6t600C0FF0000000000086DB47E331ACF00d0 /dev/did/rdisk/d19
20 scNode-A: /dev/rdisk/c6t600C0FF00000000000876E9780ECA8100d0 /dev/did/rdisk/d20
21 scNode-A: /dev/rdisk/c6t600C0FF000000000004CAD5B68A7A100d0 /dev/did/rdisk/d21
22 scNode-A: /dev/rdisk/c6t600C0FF0000000000086DB43CF85DA800d0 /dev/did/rdisk/d22
23 scNode-A: /dev/rdisk/c6t600C0FF000000000004CAD7CC3CDE500d0 /dev/did/rdisk/d23
24 scNode-A: /dev/rdisk/c6t600C0FF0000000000086DB4259B272300d0 /dev/did/rdisk/d24
25 scNode-A: /dev/rdisk/c6t600C0FF00000000000332B21D0B90000d0 /dev/did/rdisk/d25
26 scNode-A: /dev/rdisk/c6t600C0FF000000000004CAD139A855500d0 /dev/did/rdisk/d26
27 scNode-A: /dev/rdisk/c6t600C0FF00000000000332B057D2FF100d0 /dev/did/rdisk/d27
28 scNode-A: /dev/rdisk/c6t600C0FF000000000004CAD4C40941C00d0 /dev/did/rdisk/d28
```

The mirroring scheme is as follows:

```
21 <-> 13
14 <-> 17
23 <-> 16
15 <-> 19
```

4. Add devices to the set on one node.

For example:

```
# metaset -s datadg -a /dev/did/rdisk/d21 /dev/did/rdisk/d13
/dev/did/rdisk/d14 \
/dev/did/rdisk/d17 /dev/did/rdisk/d23 /dev/did/rdisk/d16
/dev/did/rdisk/d15 \
/dev/did/rdisk/d19
```

5. Create the mirrors on one node.

For example:

```
metainit -s datadg d10 1 1 /dev/did/dsk/d21s0
metainit -s datadg d11 1 1 /dev/did/dsk/d13s0
metainit -s datadg d1 -m d10
metattach -s datadg d11 d1

metainit -s datadg d20 1 1 /dev/did/dsk/d14s0
metainit -s datadg d21 1 1 /dev/did/dsk/d17s0
metainit -s datadg d2 -m d20
metattach -s datadg d21 d2

metainit -s datadg d30 1 1 /dev/did/dsk/d23s0
metainit -s datadg d31 1 1 /dev/did/dsk/d16s0
metainit -s datadg d3 -m d30
metattach -s datadg d31 d3

metainit -s datadg d40 1 1 /dev/did/dsk/d15s0
metainit -s datadg d41 1 1 /dev/did/dsk/d19s0
metainit -s datadg d4 -m d40
metattach -s datadg d41 d4

metainit -s datadg d51 -p d1 10m
metainit -s datadg d52 -p d1 200m
metainit -s datadg d53 -p d1 800m

metainit -s datadg d61 -p d2 10m
metainit -s datadg d62 -p d2 200m
metainit -s datadg d63 -p d2 800m

metainit -s datadg d71 -p d1 500m
metainit -s datadg d72 -p d1 65g

metainit -s datadg d81 -p d2 500m
metainit -s datadg d82 -p d2 65g
```

6. Perform the Sun StorageTek QFS installation on each node.

For example:

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

7. Create the mcf file on each node.

For example:

```
/etc/opt/SUNWsamfs/mcf file:

#
# File system Data
#
Data                2    ma   Data   on    shared
/dev/md/datadg/dsk/d53 20   mm   Data   on
/dev/md/datadg/dsk/d63 21   mm   Data   on
/dev/md/datadg/dsk/d3  22   mr   Data   on
/dev/md/datadg/dsk/d4  23   mr   Data   on
#

# File system Crs
#
Crs                  4    ma   Crs    on    shared
/dev/md/datadg/dsk/d51 40   mm   Crs    on
/dev/md/datadg/dsk/d61 41   mm   Crs    on
/dev/md/datadg/dsk/d52 42   mr   Crs    on
/dev/md/datadg/dsk/d62 43   mr   Crs    on
#

# File system Redo
#
Redo                 6    ma   Redo   on    shared
/dev/md/datadg/dsk/d71 60   mm   Redo   on
/dev/md/datadg/dsk/d81 61   mm   Redo   on
/dev/md/datadg/dsk/d72 62   mr   Redo   on
/dev/md/datadg/dsk/d82 63   mr   Redo   on
```

8. Create the file system hosts files.

For example:

```
/etc/opt/SUNWsamfs/hosts.Data
/etc/opt/SUNWsamfs/hosts.Crs
/etc/opt/SUNWsamfs/hosts.Oracle

# scNode-A:root> /usr/cluster/bin/scconf -p |egrep "Cluster node
name:|Node private hostname:"
      Cluster node name:      scNode-A
      Node private hostname:  clusternode1-priv
      Cluster node name:      scNode-B
      Node private hostname:  clusternode2-priv

# Host          Host IP          Server  Not   MDS Server
# Name          Address          Priority Used   Host
#-----
scNode-A       clusternode1-priv  1      -     server
scNode-B       clusternode2-priv  2      -
```

9. Create the /etc/opt/SUNWsamfs/samfs.cmd file.

For example:

```
fs = Data
    stripe=1
    sync_meta=1
    mh_write
    qwrite
    forcedirectio
    notrace
    rdlease=300
    wrlease=300
    aplease=300
```

```

fs = Crs
    stripe=1
    sync_meta=1
    mh_write
    qwrite
    forcedirectio
    notrace
    rdlease=300
    wrlease=300
    aplease=300

fs = Redo
    stripe=1
    sync_meta=1
    mh_write
    qwrite
    forcedirectio
    notrace
    rdlease=300
    wrlease=300
    aplease=300

```

10. **Create the Sun StorageTek QFS file systems. See the *Sun StorageTek QFS Installation and Upgrade Guide* for more information.**

For example:

```

/opt/SUNWsamfs/sbin/sammkfs -S <filesystem>

```

11. **Configure the resource group in Sun Cluster to manage failover of the Sun StorageTek QFS metadata server.**

- a. **Build and append the `/etc/vfstab` mount entries.**

For example:

```

#
#
# RAC on shared QFS
Data - /cluster/Data samfs - no shared,notrace
Redo - /cluster/Redo samfs - no shared,notrace
Crs - /cluster/Crs samfs - no shared,notrace

```

b. Mount the file systems across the cluster on each node.

First, mount the shared Sun StorageTek QFS file systems on the current metadata server, and then mount the file system on each metadata client.

To verify this step, type:

```
# df -h -F samfs
```

c. Create the Sun Cluster resource group to manage the metadata server.

Register the QFS resource type:

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

Add the resource group with the Sun Cluster and shared Sun StorageTek QFS metadata nodes:

```
# scrgadm -a -g sc-QFS-rg -h scNode-A,sc-Node-B -y  
RG_DEPENDENCIES="rac-framework-rg"
```

Add the shared Sun StorageTek QFS file system resource and the SUNWqfs resource type to the resource group:

```
# scrgadm -a -g sc-QFS-rg -t SUNW.qfs -j sc-qfs-fs-rs -x  
QFSfileSystem=/cluster/Data, \  
/cluster/Redo,/cluster/Crs
```

Bring the resource group online:

```
# scswitch -Z -g sc-QFS-rg
```

The shared Sun StorageTek QFS file system is now ready to use.

About Configuration Examples

This chapter provides configuration examples for the Sun StorageTek QFS shared file system on a Sun Cluster system and for the unshared Sun StorageTek QFS file system on a Sun Cluster system. All configuration examples are based on a platform consisting of the following:

- A two-node SunPlex™ system (scnode-A and scnode-B)
- DID devices that contain multi-host-accessible disk devices for both high availability and redundancy

All configurations in this chapter are also based on [CODE EXAMPLE 6-1](#). In this code example, the `scdidadm(1M)` command displays the DID devices, and the `-L` option lists the DID device paths, including those on all nodes in the Sun Cluster system.

CODE EXAMPLE 6-1 Command That Lists the DID Devices and Their DID Device Paths

```
# scdidadm -L
1  scnode-A:/dev/dsk/c0t0d0      /dev/did/dsk/d1
2  scnode-A:/dev/dsk/c0t1d0      /dev/did/dsk/d2
3  scnode-A:/dev/dsk/c0t6d0      /dev/did/dsk/d3
4  scnode-A:/dev/dsk/c6t1d0      /dev/did/dsk/d4
4  scnode-B:/dev/dsk/c7t1d0      /dev/did/dsk/d4
5  scnode-A:/dev/dsk/c6t2d0      /dev/did/dsk/d5
5  scnode-B:/dev/dsk/c7t2d0      /dev/did/dsk/d5
6  scnode-A:/dev/dsk/c6t3d0      /dev/did/dsk/d6
6  scnode-B:/dev/dsk/c7t3d0      /dev/did/dsk/d6
7  scnode-A:/dev/dsk/c6t4d0      /dev/did/dsk/d7
7  scnode-B:/dev/dsk/c7t4d0      /dev/did/dsk/d7
8  scnode-A:/dev/dsk/c6t5d0      /dev/did/dsk/d8
8  scnode-B:/dev/dsk/c7t5d0      /dev/did/dsk/d8
9  scnode-B:/dev/dsk/c0t6d0      /dev/did/dsk/d9
10 scnode-B:/dev/dsk/c1t0d0      /dev/did/dsk/d10
11 scnode-B:/dev/dsk/c1t1d0      /dev/did/dsk/d11
```

[CODE EXAMPLE 6-1](#) shows that DID devices d4 through d8 are accessible from both Sun Cluster systems (`scnode-A` and `scnode-B`). With the Sun StorageTek QFS file system sizing requirements and with knowledge of your intended application and configuration, you can decide on the most appropriate apportioning of devices to file systems. By using the Solaris `format(1M)` command, you can determine the sizing and partition layout of each DID device and resize the partitions on each DID device, if needed. Given the available DID devices, you can also configure multiple devices and their associated partitions to contain the file systems, according to your sizing requirements.

Configuring a Sun StorageTek QFS Shared File System in a Sun Cluster Environment

When you install a Sun StorageTek QFS shared file system in a Sun Cluster environment, you configure the file system's metadata server under the `SUNW.qfs` resource type. This makes the metadata server highly available and enables the Sun StorageTek QFS shared file system to be globally accessible on all configured nodes in the Sun Cluster environment.

A Sun StorageTek QFS shared file system is typically associated with a scalable application. The Sun StorageTek QFS shared file system is mounted on, and the scalable application is active on, one or more Sun Cluster nodes.

If a node in the Sun Cluster system fails, or if you switch over the resource group, the metadata server resource (Sun StorageTek QFS Sun Cluster agent) automatically relocates the file system's metadata server as necessary. This ensures that the other nodes' access to the shared file system is not affected.

Note – To manually relocate the metadata server for a Sun StorageTek QFS shared file system that is under control of the Sun Cluster system, you must use the Sun Cluster administrative commands. For more information about these commands, see the Sun Cluster documentation.

Metadata Server Resource Considerations

When the Sun Cluster system boots, the metadata server resource ensures that the file system is mounted on all nodes that are part of the resource group. However, the file system mount on those nodes is not monitored. Therefore, in certain failure cases, the file system might be unavailable on certain nodes, even if the metadata server resource is in the online state.

If you use Sun Cluster administrative commands to bring the metadata server resource group offline, the file system under the metadata server resource remains mounted on the nodes. To unmount the file system (with the exception of a node that is shut down), you must bring the metadata server resource group into the unmanaged state by using the appropriate Sun Cluster administrative command.

To remount the file system at a later time, you must bring the resource group into a managed state and then into an online state.

Example Configuration

This section shows an example of the Sun StorageTek QFS shared file system installed on raw DID devices with the Sun Cluster data service for Oracle Real Application Clusters. For detailed information on how to use the Sun StorageTek QFS shared file system with the Sun Cluster data service for Oracle Real Application Clusters, see the *Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS*.

As shown in [CODE EXAMPLE 6-1](#), DID devices d4 through d8 are highly available and are contained on controller-based storage. For you to configure a Sun StorageTek QFS shared file system in a Sun Cluster environment, the controller-based storage must support device redundancy by using RAID-1 or RAID-5.

For simplicity in this example, two file systems are created:

- `qfs1`—This file system is used for the Oracle Real Application Clusters shared installation, configuration, and log files.
- `qfs2`—This file system is used for the database files that are shared by the Oracle Real Application Clusters software.

Additionally, device d4 is used for Sun StorageTek QFS metadata. This device has two 50-gigabyte slices. The remaining devices, d5 through d8, are used for Sun StorageTek QFS file data.

This configuration involves five main steps, as detailed in the following subsections:

1. Preparing to create Sun StorageTek QFS file systems
2. Creating the file systems and configuring the Sun Cluster nodes
3. Validating the configuration
4. Configuring the network name service
5. Configuring the Sun Cluster data service for Oracle Real Application Clusters

▼ To Prepare to Create Sun StorageTek QFS Shared File Systems

1. From one node in the Sun Cluster system, use the `format(1M)` utility to lay out partitions on `/dev/did/dsk/d4` (CODE EXAMPLE 6-2).

In this example, the action is performed from node `scnode-A`.

CODE EXAMPLE 6-2 Laying Out Partitions on `/dev/did/dsk/d4`

```
# format /dev/did/rdisk/d4s2
# format> partition
[ output deleted ]
# partition> print
Current partition table (unnamed):
Total disk cylinders available: 12800 + 2 (reserved cylinders)

Part      Tag      Flag      Cylinders      Size      Blocks
 0        usr      wm         1 - 6400      50.00GB   (6400/0/0) 104857600
 1        usr      wm      6401 - 12800   50.00GB   (6400/0/0) 104857600
 2      backup  wu         0 - 12800     100.00GB  (6400/0/0) 209715200
 3 unassigned wu         0              0          (0/0/0)      0
 4 unassigned wu         0              0          (0/0/0)      0
 5 unassigned wu         0              0          (0/0/0)      0
 6 unassigned wu         0              0          (0/0/0)      0
 7 unassigned wu         0              0          (0/0/0)      0
```

NOTE: Partition 2 (backup) will not be used and was created by `format(1M)` by default.

Partition (or slice) 0 skips over the volume's Volume Table of Contents (VTOC) and is then configured as a 50-gigabyte partition. Partition 1 is configured to be the same size as partition 0.

2. On the same node, use the `format(1M)` utility to lay out partitions on `/dev/did/dsk/d5` (CODE EXAMPLE 6-3).

CODE EXAMPLE 6-3 Laying Out Partitions on /dev/did/dsk/d5

```
# format /dev/did/rdisk/d5s2
# format> partition
[ output deleted ]
# partition> print
Current partition table (unnamed):
Total disk cylinders available: 34530 + 2 (reserved cylinders)
```

Part	Tag	Flag	Cylinders	Size	Blocks
0	usr	wm	1 - 34529	269.77GB	(34529/0/0) 565723136
1	usr	wm	0 - 0	0	(0/0/0)
2	backup	wu	0 - 34529	269.77GB	(34530/0/0) 565739520
3	unassigned	wu	0	0	(0/0/0) 0
4	unassigned	wu	0	0	(0/0/0) 0
5	unassigned	wu	0	0	(0/0/0) 0
6	unassigned	wu	0	0	(0/0/0) 0
7	unassigned	wu	0	0	(0/0/0) 0

NOTE: Partition 2 (backup) will not be used and was created by format(1M) by default.

3. Still on the same node, replicate the device d5 partitioning to devices d6 through d8.

This example shows the command for device d6:

```
# prtvtoc /dev/did/rdisk/d5s2 | fmthard -s - /dev/did/rdisk/d6s2
```

4. On all nodes that are potential hosts of the file systems, perform the following:
 - a. Configure the six partitions into two Sun StorageTek QFS shared file systems by adding two new configuration entries (qfs1 and qfs2) to the mcf file (CODE EXAMPLE 6-4).

CODE EXAMPLE 6-4 Adding Configuration Entries to the mcf File

```
# cat >> /etc/opt/SUNWsamfs/mcf <<EOF
#
# Sun StorageTek QFS file system configurations
#
# Equipment      Equipment      Equipment      Family      Device      Additional
# Identifier      Ordinal        Type           Set         State       Parameters
# -----
qfs1              100           ma             qfs1        -           shared
/dev/did/dsk/d4s0 101           mm             qfs1        -
/dev/did/dsk/d5s0 102           mr             qfs1        -
/dev/did/dsk/d6s0 103           mr             qfs1        -

qfs2              200           ma             qfs2        -           shared
/dev/did/dsk/d4s1 201           mm             qfs2        -
/dev/did/dsk/d7s0 202           mr             qfs2        -
/dev/did/dsk/d8s0 203           mr             qfs2        -

EOF
```

For more information about the mcf file, see [“Function of the mcf File” on page 34](#) or the *Sun StorageTek QFS Installation and Upgrade Guide*.

- b. Edit the /etc/opt/SUNWsamfs/samfs.cmd file to add the mount options that are required for the Sun Cluster data service for Oracle Real Application Clusters (CODE EXAMPLE 6-5).**

CODE EXAMPLE 6-5 Example samfs.cmd File

```
fs = qfs2
stripe = 1
sync_meta = 1
mh_write
qwrite
forcedirectio
rdlease = 300
```

For more information about the mount options that are required by the Sun Cluster data service for Oracle Real Application Clusters, see the *Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS*.

c. **Validate that the configuration is correct.**

Be sure to perform this validation after you have configured the `mcf` file and the `samfs.cmd` file on each node.

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

▼ **To Create the Sun StorageTek QFS Shared File System and Configure Sun Cluster Nodes**

Perform this procedure for each file system you are creating. This example describes how to create the `qfs1` file system.

1. **Obtain the Sun Cluster private interconnect names by using the following command:**

```
# /usr/cluster/bin/scconf -p | egrep "Cluster node name:|Node private \
hostname:"
Cluster node name:                scnode-A
Node private hostname:            clusternode1-priv
Cluster node name:                scnode-B
Node private hostname:            clusternode2-priv
```

2. **On each node that is a potential host of the file system, do the following:**

a. **Use the `samd(1M) config` command, which signals to the Sun StorageTek QFS daemon that a new Sun StorageTek QFS configuration is available:**

```
# samd config
```

b. **Create the Sun StorageTek QFS shared hosts file for the file system (`/etc/opt/SUNWsamfs/hosts.family-set-name`), based on the Sun Cluster system's private interconnect names that you obtained in Step 1.**

3. **Edit the unique Sun StorageTek QFS shared file system's host configuration file with the Sun Cluster system's interconnect names (CODE EXAMPLE 6-6).**

For Sun Cluster software failover and fencing operations, the Sun StorageTek QFS shared file system must use the same interconnect names as the Sun Cluster system.

CODE EXAMPLE 6-6 Editing Each File System's Host Configuration File

```
# cat > hosts.qfs1 <<EOF
# File /etc/opt/SUNWsamfs/hosts.qfs1
# Host      Host IP                      Server  Not  Server
# Name      Addresses                          Priority Used Host
# -----
scnode-A    clusternode1-priv                    1      -   server
scnode-B    clusternode2-priv                    2      -
EOF
```

4. From one node in the Sun Cluster system, use the `sammkfs(1M) -S` command to create the Sun StorageTek QFS shared file system:

```
# sammkfs -S qfs1 < /dev/null
```

5. On each node that is a potential host of the file system, do the following:

- a. Use the `mkdir(1M)` command to create a global mount point for the file system, use the `chmod(1M)` command to make `root` the owner of the mount point, and use the `chown(1M)` command to make the mount point usable by `other` with read/write (755) access:

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

- b. Add the Sun StorageTek QFS shared file system entry to the `/etc/vfstab` file:

```
# cat >> /etc/vfstab <<EOF
# device      device      mount      FS      fsck      mount      mount
# to mount    to fsck     point      type     pass     at boot    options
#
qfs1          -          /global/qfs1  samfs   -         no         shared
EOF
```

▼ To Validate the Configuration

Perform this procedure for each file system you create. This example describes how to validate the configuration for file system `qfs1`.

1. If you do not know which node is acting as the metadata server for the file system, use the `samsharefs(1M) -R` command.

In [CODE EXAMPLE 6-7](#) the metadata server for `qfs1` is `scnode-A`.

CODE EXAMPLE 6-7 Determining Which Node Is the Metadata Server

```
# samsharefs -R qfs1
#
# Host file for family set 'qfs1'
#
# Version: 4      Generation: 1      Count: 2
# Server = host 1/scnode-A, length = 165
#
scnode-A clusternode2-priv 1 - server
scnode-B clusternode2-priv 2 -
```

2. Use the `mount(1M)` command to mount the file system first on the metadata server and then on each node in the Sun Cluster system.

Note – It is important that you mount the file system on the metadata server first.

```
# mount qfs1
# ls /global/qfs1
lost+found/
```

3. Validate voluntary failover by issuing the `samsharefs(1M) -s` command, which changes the Sun StorageTek QFS shared file system between nodes:

```
# samsharefs -s scnode-B qfs1
# ls /global/qfs1
lost+found/
# samsharefs -s scnode-A qfs1
# ls /global/qfs1
lost+found
```

4. Validate that the required Sun Cluster resource type is added to the resource configuration:

```
# scrgadm -p | egrep "SUNW.qfs"
```

5. If you cannot find the Sun Cluster resource type, use the `scrgadm(1M) -a -t` command to add it to the resource configuration:

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

6. Register and configure the `SUNW.qfs` resource type:

```
# scrgadm -a -g qfs-rg -h scnode-A,scnode-B
# scrgadm -a -g qfs-rg -t SUNW.qfs -j qfs-res \
-x QFSFileSystem=/global/qfs1,/global/qfs2
```

7. Use the `scswitch(1M) -z -g` command to bring the resource group online:

```
# scswitch -z -g qfs-rg
```

8. Ensure that the resource group is functional on all configured nodes:

```
# scswitch -z -g qfs-rg -h scnode-B
# scswitch -z -g qfs-rg -h scnode-A
```

▼ To Configure the Sun Cluster Data Service for Oracle Real Application Clusters

This section provides an example of how to configure the data service for Oracle Real Application Clusters for use with Sun StorageTek QFS shared file systems. For more information, see the *Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS*.

1. Install the data service as described in the *Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS*.
2. Mount the Sun StorageTek QFS shared file systems.
3. Set the correct ownership and permissions on the file systems so that the Oracle database operations are successful:

```
# chown oracle:dba /global/qfs1 /global/qfs2
# chmod 755 /global/qfs1 /global/qfs2
```

4. As the `oracle` user, create the subdirectories that are required for the Oracle Real Application Clusters installation and database files:

```
$ id
uid=120(oracle) gid=520(dba)
$ mkdir /global/qfs1/oracle_install
$ mkdir /global/qfs2/oracle_db
```

The Oracle Real Application Clusters installation uses the `/global/qfs1/oracle_install` directory path as the value for the `ORACLE_HOME` environment variable that is used in Oracle operations. The Oracle Real Application Clusters database files' path is prefixed with the `/global/qfs2/oracle_db` directory path.

5. Install the Oracle Real Application Clusters software.

During the installation, provide the path for the installation defined in Step 4 (`/global/qfs1/oracle_install`).

6. Create the Oracle Real Application Clusters database.

During database creation, specify that you want the database files located in the `qfs2` shared file system.

7. If you are automating the startup and shutdown of Oracle Real Application Clusters database instances, ensure that the required dependencies for resource groups and resources are set.

For more information, see the *Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS*.

Note – If you plan to automate the startup and shutdown of Oracle Real Application Clusters database instances, you must use Sun Cluster software version 3.1 9/04 or a compatible version.

Note – In shared Sun StorageTek QFS configurations that are being used for Oracle RAC 10g configurations, when the Oracle installer for Cluster Ready Services (CRS) prompts the user to execute `root.sh` this command fails in some instances. In other instances, when an Oracle Cluster Registry (OCR) file is created by `root.sh`, it makes the CRS registry unstable.

The workaround is to preallocate the OCR file to be larger than 700416. For example, preallocate a one-megabyte file, as user `oracle`, before running `root.sh`, as shown here:

```
$ dd if=/dev/zero of=<OCR file path> bs=1024k count=1M
```

Configuring an Unshared File System in a Sun Cluster Environment

When you install the unshared Sun StorageTek QFS file system on a Sun Cluster system, you configure the file system for high availability (HA) under the Sun Cluster `HASStoragePlus` resource type. An unshared Sun StorageTek QFS file system in a Sun Cluster system is typically associated with one or more failover applications, such as highly available network file server (HA-NFS) or highly available ORACLE (HA-ORACLE). Both the unshared Sun StorageTek QFS file system and the failover applications are active in a single resource group; the resource group is active on one Sun Cluster node at a time.

An unshared Sun StorageTek QFS file system is mounted on a single node at any given time. If the Sun Cluster fault monitor detects an error, or if you switch over the resource group, the unshared Sun StorageTek QFS file system and its associated HA applications fail over to another node, depending on how the resource group has been previously configured.

Any file system contained on a Sun Cluster global device group (`/dev/global/*`) can be used with the `HASStoragePlus` resource type. When a file system is configured with the `HASStoragePlus` resource type, it becomes part of a Sun Cluster resource group and the file system under Sun Cluster Resource Group Manager (RGM) control is mounted locally on the node where the resource group is active. When the RGM causes a resource group switchover or fails over to another configured Sun Cluster node, the unshared Sun StorageTek QFS file system is unmounted from the current node and remounted on the new node.

Each unshared Sun StorageTek QFS file system requires a minimum of two raw disk partitions or volume manager-controlled volumes (Solstice DiskSuite™/Solaris Volume Manager or VERITAS Volume Manager), one for Sun StorageTek QFS metadata (inodes) and one for Sun StorageTek QFS file data. Configuring multiple partitions or volumes across multiple disks through multiple data paths increases unshared Sun StorageTek QFS file system performance. For information about sizing metadata and file data partitions, see [“Design Basics” on page 7](#).

This section provides three examples of Sun Cluster system configurations using the unshared Sun StorageTek QFS file system. In these examples, a file system is configured in combination with an HA-NFS file mount point on the following:

- Raw global devices in Example 1
- Solstice DiskSuite/Solaris Volume Manager-controlled volumes in Example 2
- VERITAS Volume Manager (VxVM)-controlled volumes in Example 3

For simplicity in all of these configurations, ten percent of each file system is used for Sun StorageTek QFS metadata, and the remaining space is used for Sun StorageTek QFS file data. For information about sizing and disk layout considerations, see the *Sun StorageTek QFS Installation and Upgrade Guide*.

Example 1: HA-NFS on Raw Global Devices

This example shows how to configure the unshared Sun StorageTek QFS file system with HA-NFS on raw global devices. For this configuration, the raw global devices must be contained on controller-based storage. This controller-based storage must support device redundancy through RAID-1 or RAID-5.

As shown in [CODE EXAMPLE 6-1](#), the DID devices used in this example, `d4` through `d7`, are highly available and are contained on controller-based storage. The `HAStoragePlus` resource type requires the use of global devices, so each DID device (`/dev/did/dsk/dx`) is accessible as a global device by using the following syntax: `/dev/global/dsk/dx`.

The main steps in this example are as follows:

1. Prepare to create an unshared file system.
2. Create the file system and configure the Sun Cluster nodes.
3. Configure the network name service and the IP Measurement Protocol (IPMP) validation testing.
4. Configure HA-NFS and configure the file system for high availability.

▼ To Prepare to Create an Unshared Sun StorageTek QFS File System

1. Use the `format(1M)` utility to lay out the partitions on `/dev/global/dsk/d4`:

```
# format /dev/global/rdisk/d4s2
# format> partition
[ output deleted ]
# partition> print
Current partition table (original):
Total disk cylinders available: 34530 + 2 (reserved cylinders)
Part      Tag      Flag      Cylinders      Size      Blocks
0  unassigned  wm        1 - 3543      20.76GB   (3543/0/0)  43536384
1  unassigned  wm      3544 - 34529  181.56GB  (30986/0/0) 380755968
2  backup      wu         0 - 34529  202.32GB  (34530/0/0) 424304640
3  unassigned  wu         0              0          (0/0/0)      0
4  unassigned  wu         0              0          (0/0/0)      0
5  unassigned  wu         0              0          (0/0/0)      0
6  unassigned  wu         0              0          (0/0/0)      0
7  unassigned  wu         0              0          (0/0/0)      0

NOTE: Partition 2 (backup) will not be used and was created by format(1m) by
default.
```

Partition (or slice) 0 skips over the volume's Volume Table of Contents (VTOC) and is then configured as a 20-gigabyte partition. The remaining space is configured into partition 1.

2. Replicate the global device `d4` partitioning to global devices `d5` through `d7`.

This example shows the command for global device `d5`:

```
# prtvtoc /dev/global/rdisk/d4s2 | fmthard \
-s - /dev/global/rdisk/d5s2
```

3. On all nodes that are potential hosts of the file system, perform the following:

- a. Configure the eight partitions (four global devices, with two partitions each) into a Sun StorageTek QFS file system by adding a new file system entry to the `mcf` file.

```
# cat >> /etc/opt/SUNWsamfs/mcf <<EOF
#
# Sun StorageTek QFS file system configurations
#
# Equipment      Equipment      Equipment      Family      Device      Additional
# Identifier      Ordinal        Type           Set         State       Parameters
# -----
qfsnfs1          100            ma             qfsnfs1     on
/dev/global/dsk/d4s0 101            mm             qfsnfs1
/dev/global/dsk/d5s0 102            mm             qfsnfs1
/dev/global/dsk/d6s0 103            mm             qfsnfs1
/dev/global/dsk/d7s0 104            mm             qfsnfs1
/dev/global/dsk/d4s1 105            mr             qfsnfs1
/dev/global/dsk/d5s1 106            mr             qfsnfs1
/dev/global/dsk/d6s1 107            mr             qfsnfs1
/dev/global/dsk/d7s1 108            mr             qfsnfs1
EOF
```

For information about the `mcf` file, see [“Function of the `mcf` File”](#) on page 34.

- b. Validate that the configuration information you added to the `mcf` file is correct, and fix any errors in the `mcf` file before proceeding.

It is important to complete this step before you configure the Sun StorageTek QFS file system under the `HASStoragePlus` resource type.

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

▼ To Create the Sun StorageTek QFS File System and Configure Sun Cluster Nodes

1. On each node that is a potential host of the file system, issue the `samd(1M)` `config` command.

This command signals to the Sun StorageTek QFS daemon that a new Sun StorageTek QFS configuration is available.

```
# samd config
```

2. From one node in the Sun Cluster system, use the `sammkfs(1M)` command to create the file system:

```
# sammkfs qfsnfs1 < /dev/null
```

3. On each node that is a potential host of the file system, do the following:
 - a. Use the `mkdir(1M)` command to create a global mount point for the file system, use the `chmod(1M)` command to make `root` the owner of the mount point, and use the `chown(1M)` command to make the mount point usable by `other` with read/write (755) access:

```
# mkdir /global/qfsnfs1
# chmod 755 /global/qfsnfs1
# chown root:other /global/qfsnfs1
```

- b. Add the Sun StorageTek QFS file system entry to the `/etc/vfstab` file.

Note that the mount options field contains the `sync_meta=1` value.

```
# cat >> /etc/vfstab <<EOF
# device      device      mount      FS      fsck      mount      mount
# to mount    to fsck     point      type     pass      at boot    options
#
qfsnfs1      -          /global/qfsnfs1  samfs    2         no         sync_meta=1
EOF
```

- c. Validate the configuration by mounting and unmounting the file system:

```
# mount qfsnfs1
# ls /global/qfsnfs1
lost+found/
# umount qfsnfs1
```

4. Use the `scrgadm(1M) -p | egrep` command to validate that the required Sun Cluster resource types have been added to the resource configuration:

```
# scrgadm -p | egrep "SUNW.HAStoragePlus|SUNW.LogicalHostname|SUNW.nfs"
```

5. If you cannot find a required Sun Cluster resource type, use the `scrgadm(1M) -a -t` command to add it to the configuration:

```
# scrgadm -a -t SUNW.HAStoragePlus
# scrgadm -a -t SUNW.LogicalHostname
# scrgadm -a -t SUNW.nfs
```

▼ To Configure the Network Name Service and the IPMP Validation Testing

This section provides an example of how to configure the network name service and the IPMP Validation Testing for your Sun Cluster nodes. For more information, see the *Sun Cluster Software Installation Guide for Solaris OS*, the *System Administration Guide: IP Services*, and the *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

1. Use `vi` or another text editor to edit the `/etc/nsswitch.conf` file so that it looks in the Sun Cluster system and files for node names.

Perform this step before you configure the Network Information Name service (NIS) server.

```

# cat /etc/nsswitch.conf
#
# /etc/nsswitch.nis:
#
# An example file that could be copied over to /etc/nsswitch.conf; it
# uses NIS (YP) in conjunction with files.
#
# the following two lines obviate the "+" entry in /etc/passwd and /etc/group.
passwd:    files nis
group:     files nis

# Cluster s/w and local /etc/hosts file take precedence over NIS
hosts:    cluster files nis [NOTFOUND=return]
ipnodes:  files
# Uncomment the following line and comment out the above to resolve
# both IPv4 and IPv6 addresses from the ipnodes databases. Note that
# IPv4 addresses are searched in all of the ipnodes databases before
# searching the hosts databases. Before turning this option on, consult
# the Network Administration Guide for more details on using IPv6.
# ipnodes: nis [NOTFOUND=return] files

networks: nis[NOTFOUND=return] files
protocols: nis [NOTFOUND=return] files
rpc: nis[NOTFOUND=return] files
ethers: nis[NOTFOUND=return] files
netmasks: nis[NOTFOUND=return] files
bootparams: nis[NOTFOUND=return] files
publickey: nis[NOTFOUND=return] files

netgroup: nis

automount: files nis
aliases: files nis
[remainder of file content not shown]

```

2. Verify that the changes you made to the /etc/nsswitch.conf are correct:

```

# grep '^hosts:' /etc/nsswitch.conf
hosts:    cluster files nis [NOTFOUND=return]
#

```

3. Set up IPMP validation testing using available network adapters.

The adapters qfe2 and qfe3 are used as examples.

a. Statically configure the IPMP test address for each adapter:

```
#cat >> /etc/hosts << EOF
#
# Test addresses for scnode-A
#
192.168.2.2      `uname -n'-qfe2
192.168.2.3      `uname -n'-qfe2-test
192.168.3.2      `uname -n'-qfe3
192.168.3.3      `uname -n'-qfe3-test
#
# Test addresses for scnode-B
#
192.168.2.4      `uname -n'-qfe2
192.168.2.5      `uname -n'-qfe2-test
192.168.3.4      `uname -n'-qfe3
192.168.3.5      `uname -n'-qfe3-test
EOF
```

b. Dynamically configure the IPMP adapters:

```
# ifconfig qfe2 plumb `uname -n'-qfe2-test netmask + broadcast + deprecated \
-failover -standby group ipmp0 up
# ifconfig qfe2 addif `uname -n'-qfe2 up
# ifconfig qfe3 plumb `uname -n'-qfe3-test netmask + broadcast + deprecated \
-failover -standby group ipmp0 up
# ifconfig qfe3 addif `uname -n'-qfe3 up
```

c. Verify the configuration:

```
# cat > /etc/hostname.qfe2 << EOF
`uname -n'-qfe2-test netmask + broadcast + deprecated -failover -standby \
group ipmp0 up addif `uname -n'-qfe2 up
EOF

# cat > /etc/hostname.qfe3 << EOF
`uname -n'-qfe3-test netmask + broadcast + deprecated -failover -standby \
group ipmp0 up addif `uname -n'-qfe3 up
EOF
```

▼ To Configure HA-NFS and the Sun StorageTek QFS File System for High Availability

This section provides an example of how to configure HA-NFS. For more information about HA-NFS, see the *Sun Cluster Data Service for Network File System (NFS) Guide for Solaris OS* and your NFS documentation.

1. Create the NFS share point for the Sun StorageTek QFS file system.

Note that the share point is contained within the `/global` file system, not within the Sun StorageTek QFS file system.

```
# mkdir -p /global/nfs/SUNW.nfs
# echo "share -F nfs -o rw /global/qfsnfs1" > \
/global/nfs/SUNW.nfs/dfstab.nfs1-res
```

2. Create the NFS resource group:

```
# scrgadm -a -g nfs-rg -y PathPrefix=/global/nfs
```

3. Add the NFS logical host to the `/etc/hosts` table, using the address for your site:

```
# cat >> /etc/hosts << EOF
#
# IP Addresses for LogicalHostnames
#
192.168.2.10      lh-qfs1
EOF
```

4. Use the `scrgadm(1M) -a -L -g` command to add the logical host to the NFS resource group:

```
# scrgadm -a -L -g nfs-rg -l lh-nfs1
```

5. Use the `scrgadm(1M) -c -g` command to configure the `HASStoragePlus` resource type:

```
# scrgadm -c -g nfs-rg -h scnode-A,scnode-B
# scrgadm -a -g nfs-rg -j qfsnfs1-res -t SUNW.HASStoragePlus \
-x FilesystemMountPoints=/global/qfsnfs1 \
-x FilesystemCheckCommand=/bin/true
```

6. Bring the resource group online:

```
# scswitch -Z -g nfs-rg
```

7. Configure the NFS resource type and set a dependency on the `HAStoragePlus` resource:

```
# scrgadm -a -g nfs-rg -j nfs1-res -t SUNW.nfs -y \  
Resource_dependencies=qfsnfs1-res
```

8. Bring the NFS resource online:

```
# scswitch -e -j nfs1-res
```

The NFS resource `/net/lh-nfs1/global/qfsnfs1` is now fully configured and is also highly available.

9. Before announcing the availability of the highly available NFS file system on the Sun StorageTek QFS file system, test the resource group to ensure that it can be switched between all configured nodes without errors and can be taken online and offline:

```
# scswitch -z -g nfs-rg -h scnode-A  
# scswitch -z -g nfs-rg -h scnode-B  
# scswitch -F -g nfs-rg  
# scswitch -Z -g nfs-rg
```

Example 2: HA-NFS on Volumes Controlled by Solstice DiskSuite/Solaris Volume Manager

This example shows how to configure the unshared Sun StorageTek QFS file system with HA-NFS on volumes controlled by Solstice DiskSuite/Solaris Volume Manager software. With this configuration, you can choose whether the DID devices are contained on redundant controller-based storage using RAID-1 or RAID-5 volumes. Typically, Solaris Volume Manager is used only when the underlying controller-based storage is not redundant.

As shown in [CODE EXAMPLE 6-1](#), the DID devices used in this example, `d4` through `d7`, are highly available and are contained on controller-based storage. Solaris Volume Manager requires that DID devices be used to populate the raw devices from which Solaris Volume Manager can configure volumes. Solaris Volume Manager creates globally accessible disk groups, which can then be used by the `HAStoragePlus` resource type for creating Sun StorageTek QFS file systems.

This example follows these steps:

1. Prepare the Solstice DiskSuite/Solaris Volume Manager software.
2. Prepare to create an unshared file system.
3. Create the file system and configure the Sun Cluster nodes.
4. Configure the network name service and the IPMP validation testing.
5. Configure HA-NFS and configure the file system for high availability.

▼ To Prepare the Solstice DiskSuite/Solaris Volume Manager Software

1. **Determine whether a Solaris Volume Manager metadata database (`metadb`) is already configured on each node that is a potential host of the Sun StorageTek QFS file system:**

```
# metadb
```

	flags	first blk	block count	
a	m p luo	16	8192	/dev/dsk/c0t0d0s7
a	p luo	16	8192	/dev/dsk/c1t0d0s7
a	p luo	16	8192	/dev/dsk/c2t0d0s7

If the `metadb(1M)` command does not return a metadata database configuration, then on each node, create three or more database replicas on one or more local disks. Each replica must be at least 16 megabytes in size. For more information about creating the metadata database configuration, see the *Sun Cluster Software Installation Guide for Solaris OS*.

2. **Create an HA-NFS disk group to contain all Solaris Volume Manager volumes for this Sun StorageTek QFS file system:**

```
# metaset -s nfsdg -a -h scnode-A scnode-B
```

3. **Add DID devices `d4` through `d7` to the pool of raw devices from which Solaris Volume Manager can create volumes:**

```
# metaset -s nfsdg -a /dev/did/dsk/d4 /dev/did/dsk/d5 \  
/dev/did/dsk/d6 /dev/did/dsk/d7
```

▼ To Prepare For a Sun StorageTek QFS File System

1. Use the `format(1M)` utility to lay out partitions on `/dev/global/dsk/d4`:

```
# format /dev/global/rdisk/d4s2
# format> partition
[ output deleted ]
# partition> print
Current partition table (original):
Total disk cylinders available: 34530 + 2 (reserved cylinders)
Part      Tag      Flag      Cylinders      Size      Blocks
0  unassigned  wm      1 - 3543      20.76GB   (3543/0/0)  43536384
1  unassigned  wm    3544 - 34529  181.56GB   (30986/0/0) 380755968
2  backup      wu      0 - 34529    202.32GB   (34530/0/0) 424304640
3  unassigned  wu      0              0          (0/0/0)      0
4  unassigned  wu      0              0          (0/0/0)      0
5  unassigned  wu      0              0          (0/0/0)      0
6  unassigned  wu      0              0          (0/0/0)      0
7  unassigned  wu      0              0          (0/0/0)      0

NOTE: Partition 2 (backup) will not be used and was created by format(1m) by
default.
```

This example shows that partition or slice 0 skips over the volume's Volume Table of Contents (VTOC) and is then configured as a 20-gigabyte partition. The remaining space is configured into partition 1.

2. Replicate the partitioning of DID device `d4` to DID devices `d5` through `d7`.

This example shows the command for device `d5`:

```
# prtvtoc /dev/global/rdisk/d4s2 | fmthard \
-s - /dev/global/rdisk/d5s2
```

3. Configure the eight partitions (four DID devices, two partitions each) into two RAID-1 (mirrored) Sun StorageTek QFS metadata volumes and two RAID-5 (parity-striped) Sun StorageTek QFS file data volumes:

- a. Combine partition (slice) 0 of these four drives into two RAID-1 sets:

```
# metainit -s nfsdg -f d1 1 1 /dev/did/dsk/d4s0
# metainit -s nfsdg -f d2 1 1 /dev/did/dsk/d5s0
# metainit -s nfsdg d10 -m d1 d2
# metainit -s nfsdg -f d3 1 1 /dev/did/dsk/d6s0
# metainit -s nfsdg -f d4 1 1 /dev/did/dsk/d7s0
# metainit -s nfsdg d11 -m d3 d4
```

b. Combine partition 1 of these four drives into two RAID-5 sets:

```
# metainit -s nfsdg d20 -p /dev/did/dsk/d4s1 205848574b
# metainit -s nfsdg d21 -p /dev/did/dsk/d5s1 205848574b
# metainit -s nfsdg d22 -p /dev/did/dsk/d6s1 205848574b
# metainit -s nfsdg d23 -p /dev/did/dsk/d7s1 205848574b
# metainit -s nfsdg d30 -r d20 d21 d22 d23
```

c. On each node that is a potential host of the file system, add the Sun StorageTek QFS file system entry to the `mcf` file:

```
# cat >> /etc/opt/SUNWsamfs/mcf <<EOF

# Sun StorageTek QFS file system configurations
#
# Equipment      Equipment      Equipment      Family      Device      Additional
# Identifier      Ordinal        Type           Set         State       Parameters
# -----
qfsnfs1          100            ma             qfsnfs1    on
/dev/md/nfsdg/dsk/d10  101            mm             qfsnfs1
/dev/md/nfsdg/dsk/d11  102            mm             qfsnfs1
/dev/md/nfsdg/dsk/d30  103            mr             qfsnfs1
EOF
```

For more information about the `mcf` file, see [“Function of the `mcf` File” on page 34](#).

4. Validate that the `mcf(4)` configuration is correct on each node, and fix any errors in the `mcf` file before proceeding.

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

▼ To Create the Sun StorageTek QFS File System and Configure Sun Cluster Nodes

1. On each node that is a potential host of the file system, use the `samd(1M)` config command.

This command signals to the Sun StorageTek QFS daemon that a new Sun StorageTek QFS configuration is available.

```
# samd config
```

2. Enable Solaris Volume Manager mediation detection of disk groups, which assists the Sun Cluster system in the detection of drive errors:

```
# metaset -s nfsdg -a -m scnode-A
# metaset -s nfsdg -a -m scnode-B
```

3. On each node that is a potential host of the file system, ensure that the NFS disk group exists:

```
# metaset -s nfsdg -t
```

4. From one node in the Sun Cluster system, use the `sammkfs(1M)` command to create the file system:

```
# sammkfs qfsnfs1 < /dev/null
```

5. On each node that is a potential host of the file system, do the following:

- a. Use the `mkdir(1M)` command to create a global mount point for the file system, use the `chmod(1M)` command to make `root` the owner of the mount point, and use the `chown(1M)` command to make the mount point usable by `other` with read/write (755) access:

```
# mkdir /global/qfsnfs1
# chmod 755 /global/qfsnfs1
# chown root:other /global/qfsnfs1
```

- b. Add the Sun StorageTek QFS file system entry to the `/etc/vfstab` file.

Note that the mount options field contains the `sync_meta=1` value.

```
# cat >> /etc/vfstab << EOF
# device          device          mount          FS          fsck          mount          mount
# to mount        to fsck        point          type        pass          at boot        options
#
qfsnfs1          -          /global/qfsnfs1  samfs       2             no             sync_meta=1
EOF
```

- c. Validate the configuration by mounting and unmounting the file system.

Perform this step one node at a time. In this example, the `qfsnfs1` file system is

mounted and unmounted on one node.

```
# mount qfsnfs1
# ls /global/qfsnfs1
lost+found/
# umount qfsnfs1
```

Note – When testing the mount point, use the `metaset -r` (release) and `-t` (take) command to move the `nfsdg` disk group between Sun Cluster nodes. Then use the `samd(1M) config` command to alert the daemon of the configuration changes.

6. Use the `scrgadm(1M) -p | egrep` command to validate that the required Sun Cluster resource types have been added to the resource configuration:

```
# scrgadm -p | egrep "SUNW.HAStoragePlus|SUNW.LogicalHostname|SUNW.nfs"
```

7. If you cannot find a required Sun Cluster resource type, add it with one or more of the following commands:

```
# scrgadm -a -t SUNW.HAStoragePlus
# scrgadm -a -t SUNW.LogicalHostname
# scrgadm -a -t SUNW.nfs
```

▼ To Configure the Network Name Service and the IPMP Validation Testing

To configure the Network Name Service and the IPMP validation testing, follow the instructions in [“To Configure the Network Name Service and the IPMP Validation Testing” on page 149](#)

▼ To Configure HA-NFS and the Sun StorageTek QFS File System for High Availability

This section provides an example of how to configure HA-NFS. For more information about HA-NFS, see the *Sun Cluster Data Service for Network File System (NFS) Guide for Solaris OS* and your NFS documentation.

1. Create the NFS share point for the Sun StorageTek QFS file system.

Note that the share point is contained within the /global file system, not within the Sun StorageTek QFS file system.

```
# mkdir -p /global/nfs/SUNW.nfs
# echo "share -F nfs -o rw /global/qfsnfs1" > \
/global/nfs/SUNW.nfs/dfstab.nfs1-res
```

2. Create the NFS resource group:

```
# scrgadm -a -g nfs-rg -y PathPrefix=/global/nfs
```

3. Add a logical host to the NFS resource group:

```
# scrgadm -a -L -g nfs-rg -l lh-nfs1
```

4. Configure the HAStoragePlus resource type:

```
# scrgadm -c -g nfs-rg -h scnode-A,scnode-B
# scrgadm -a -g nfs-rg -j qfsnfs1-res -t SUNW.HAStoragePlus \
-x FilesystemMountPoints=/global/qfsnfs1 \
-x FilesystemCheckCommand=/bin/true
```

5. Bring the resource group online:

```
# scswitch -Z -g nfs-rg
```

6. Configure the NFS resource type and set a dependency on the HAStoragePlus resource:

```
# scrgadm -a -g nfs-rg -j nfs1-res -t SUNW.nfs -y \
Resource_dependencies=qfsnfs1-res
```

7. Use the scswitch(1M) -e -j command to bring the NFS resource online:

```
# scswitch -e -j nfs1-res
```

The NFS resource /net/lh-nfs1/global/qfsnfs1 is fully configured and highly available.

8. Before you announce the availability of the highly available NFS file system on the Sun StorageTek QFS file system, test the resource group to ensure that it can be switched between all configured nodes without errors and can be taken online and offline:

```
# scswitch -z -g nfs-rg -h scnode-A
# scswitch -z -g nfs-rg -h scnode-B
# scswitch -F -g nfs-rg
# scswitch -Z -g nfs-rg
```

Example 3: HA-NFS on VxVM Volumes

This example shows how to configure the unshared Sun StorageTek QFS file system with HA-NFS on VERITAS Volume Manager-controlled volumes (VxVM volumes). With this configuration, you can choose whether the DID devices are contained on redundant controller-based storage using RAID-1 or RAID-5. Typically, VxVM is used only when the underlying storage is not redundant.

As shown in [CODE EXAMPLE 6-1](#), the DID devices used in this example, d4 through d7, are highly available and are contained on controller-based storage. VxVM requires that shared DID devices be used to populate the raw devices from which VxVM configures volumes. VxVM creates highly available disk groups by registering the disk groups as Sun Cluster device groups. These disk groups are not globally accessible, but can be failed over, making them accessible to at least one node. The disk groups can be used by the `HASStoragePlus` resource type.

Note – The VxVM packages are separate, additional packages that must be installed, patched, and licensed. For information about installing VxVM, see the VxVM Volume Manager documentation.

To use Sun StorageTek QFS software with VxVM, you must install the following VxVM packages:

- `VRTSvlic`
- `VRTSvmdoc`
- `VRTSvmman`
- `VRTSvmpro`
- `VRTSvxvm`
- `VRTSob` and `VRTSobgui` (optional browser interface packages)

This example follows these steps:

1. Configure the VxVM software.
2. Prepare to create an unshared file system.

3. Create the file system and configure the Sun Cluster nodes.
4. Validate the configuration.
5. Configure the network name service and the IPMP validation testing.
6. Configure HA-NFS and configure the file system for high availability.

▼ To Configure the VxVM Software

This section provides an example of how to configure the VxVM software for use with the Sun StorageTek QFS software. For more detailed information about the VxVM software, see the VxVM documentation.

1. Determine the status of dynamic multipathing (DMP) for VERITAS.

```
# vxddmpadm listctlr all
```

2. Use the `scdidadm(1M)` utility to determine the HBA controller number of the physical devices to be used by VxVM.

As shown in the following example, the multi-node accessible storage is available from `scnode-A` using HBA controller `c6`, and from node `scnode-B` using controller `c7`:

```
# scdidadm -L
[ some output deleted]
4   scnode-A:/dev/dsk/c6t60020F20000037D13E26595500062F06d0 /dev/did/dsk/d4
4   scnode-B:/dev/dsk/c7t60020F20000037D13E26595500062F06d0 /dev/did/dsk/d4
```

3. Use VxVM to configure all available storage as seen through controller `c6`:

```
# vxddmpadm getsubpaths ctlr=c6
```

4. Place all of this controller's devices under VxVM control:

```
# vxddiskadd fabric_
```

5. Create a disk group, create volumes, and then start the new disk group:

```
# /usr/sbin/vxdg init qfs-dg qfs-dg00=disk0 \
qfsdg01=disk1 qfsdg02=disk2 qfsdg03=disk3
```

6. Ensure that the previously started disk group is active on this system:

```
# vxvg import nfsdg
# vxvg free
```

7. Configure two mirrored volumes for Sun StorageTek QFS metadata and two volumes for Sun StorageTek QFS file data volumes.

These mirroring operations are performed as background processes, given the length of time they take to complete.

```
# vxassist -g nfsdg make m1 10607001b
# vxassist -g nfsdg mirror m1&
# vxassist -g nfsdg make m2 10607001b
# vxassist -g nfsdg mirror m2&
# vxassist -g nfsdg make m10 201529000b
# vxassist -g nfsdg mirror m10&
# vxassist -g nfsdg make m11 201529000b
# vxassist -g nfsdg mirror m11&
```

8. Configure the previously created VxVM disk group as a Sun Cluster-controlled disk group:

```
# scconf -a -D type=vxvm,name=nfsdg,nodelist=scnode-A:scnode-B
```

▼ To Prepare to Create a Sun StorageTek QFS File System

Perform this procedure on each node that is a potential host of the file system.

1. Add the Sun StorageTek QFS file system entry to the `mcf` file.

CODE EXAMPLE 6-8 Addition of the File System to the `mcf` File

```
# cat >> /etc/opt/SUNwsamfs/mcf <<EOF
# Sun StorageTek QFS file system configurations
#
# Equipment          Equipment  Equipment  Family    Device    Additional
# Identifier         Ordinal   Type       Set       State     Parameters
# -----
qfsnfs1             100      ma        qfsnfs1   on
/dev/vx/dsk/nfsdg/m1 101      mm        qfsnfs1
/dev/vx/dsk/nfsdg/m2 102      mm        qfsnfs1
/dev/vx/dsk/nfsdg/m10 103      mr        qfsnfs1
/dev/vx/dsk/nfsdg/m11 104      mr        qfsnds1
EOF
```

For more information about the `mcf` file, see [“Function of the `mcf` File”](#) on page 34.

2. Validate that the `mcf(4)` configuration is correct, and correct any errors in the `mcf` file before proceeding:

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

▼ To Create the Sun StorageTek QFS File System and Configure Sun Cluster Nodes

1. On each node that is a potential host of the file system, use the `samd(1M)` `config` command.

This command signals to the Sun StorageTek QFS daemon that a new Sun StorageTek QFS configuration is available.

```
# samd config
```

2. From one node in the Sun Cluster system, use the `sammkfs(1M)` command to create the file system:

```
# sammkfs qfsnfs1 < /dev/null
```

3. On each node that is a potential host of the file system, do the following:

- a. Use the `mkdir(1M)` command to create a global mount point for the file system, use the `chmod(1M)` command to make `root` the owner of the mount point, and use the `chown(1M)` command to make the mount point usable by `other` with read/write (755) access:

```
# mkdir /global/qfsnfs1
# chmod 755 /global/qfsnfs1
# chown root:other /global/qfsnfs1
```

- b. Add the Sun StorageTek QFS file system entry to the `/etc/vfstab` file.

Note that the mount options field contains the `sync_meta=1` value.

```
# cat >> /etc/vfstab << EOF
# device      device      mount      FS      fsck      mount      mount
# to mount    to fsck     point      type    pass     at boot    options
#
qfsnfs1      -          /global/qfsnfs1  samfs   2         no         sync_meta=1
EOF
```

▼ To Validate the Configuration

1. Validate that all nodes that are potential hosts of the file system are configured correctly.

To do this, move the disk group that you created in [“To Configure the VxVM Software” on page 161](#) to the node, and mount and then unmount the file system. Perform this validation one node at a time.

```
# scswitch -z -D nfsdg -h scnode-B
# mount qfsnfs1
# ls /global/qfsnfs1
lost+found/
# umount qfsnfs1
```

2. Ensure that the required Sun Cluster resource types have been added to the resource configuration:

```
# scrgadm -p | egrep "SUNW.HAStoragePlus|SUNW.LogicalHostname|SUNW.nfs"
```

3. If you cannot find a required Sun Cluster resource type, add it with one or more of the following commands:

```
# scrgadm -a -t SUNW.HAStoragePlus
# scrgadm -a -t SUNW.LogicalHostname
# scrgadm -a -t SUNW.nfs
```

▼ To Configure the Network Name Service and the IPMP Validation Testing

To configure the Network Name Service and the IPMP validation testing, follow the instructions in [“To Configure the Network Name Service and the IPMP Validation Testing” on page 149](#)

▼ To Configure HA-NFS and the Sun StorageTek QFS File System for High Availability

To configure HA-NFS and the file system for high availability, follow the instructions in [“To Configure HA-NFS and the Sun StorageTek QFS File System for High Availability” on page 158](#).

Configuring Shared Clients Outside the Cluster

If you are configuring a Sun Cluster environment and would like to have shared clients that are outside of the cluster, perform the following configurations.

The example below is based on a two-node metadata server cluster configuration.

Configuration Prerequisites

The following items must be configured or verified in order to set up shared clients outside the cluster:

- [“Sun StorageTek QFS Metadata Server Sun Cluster Nodes” on page 165](#)
- [“Sun StorageTek QFS Metadata Client Nodes” on page 166](#)
- [“Sun Cluster Device Configuration” on page 166](#)
- [“Requirements for Configuring Clients Outside the Cluster” on page 166](#)
- [“Minimum Software Release Levels” on page 167](#)
- [“Hardware Architecture Supported” on page 167](#)
- [“Storage Requirements” on page 167](#)

Sun StorageTek QFS Metadata Server Sun Cluster Nodes

The following requirements must be met for the Sun StorageTek QFS metadata server Sun Cluster nodes:

- Sun StorageTek QFS metadata devices identified as `mm` devices cannot reside on the same physical LUNs with the data devices.
- The Sun Cluster `scconf localonly` flag cannot be enabled for the metadata devices defined as `mm` devices in the Shared Sun StorageTek QFS configuration.
- Sun StorageTek QFS data devices identified as `mr` devices cannot reside on the same physical LUNs with the metadata devices.

Sun StorageTek QFS Metadata Client Nodes

The following requirements must be met for Sun StorageTek QFS metadata client nodes:

- Sun StorageTek QFS metadata devices identified as `mm` devices need to be identified as `nodev` devices in the Shared Sun StorageTek QFS configuration.
- Data devices identified in as `mr` devices on the Sun StorageTek QFS clients must follow the `/dev/dsk` syntax instead of the `/dev/did/dsk` syntax used on clustered nodes.

Sun Cluster Device Configuration

The `localonly` flag must be set on all data devices. Setting the local mode for data devices via the `/etc/opt/SUNWsamfs/mcf` file identifies which devices are to be used as Sun StorageTek QFS data devices.

Perform the following as `root` on any nodes running under Sun Cluster:

```
scconf -r -D name=dsk/dX,nodelist=node2
```

```
scconf -c -D name=dsk/dX,localonly=true
```

Requirements for Configuring Clients Outside the Cluster

Due to the complexity of a configuration that includes both Sun Cluster and Shared Sun StorageTek QFS clients, a separate private network is mandatory for Sun StorageTek QFS metadata traffic. In addition, the following should also be true:

- Sun Cluster nodes used for the metadata server (MDS) server should use IPMP groups.
- Sun Cluster Sun StorageTek QFS resource groups should have a logical hostname configured for metadata communication between the MDS and the clients that exist outside the cluster.
- Sun Cluster Sun StorageTek QFS resource groups should use the dependency properties between the logical hostname resource and the Sun StorageTek QFS file system resource. For example, the Sun StorageTek QFS file system resource would be dependent on the logical hostname resource.
- Only `ma`, `mr`, and `gx` devices are supported.
- No software volume managers are supported for this configuration.
- No other Sun Cluster agents are supported.

Minimum Software Release Levels

The following minimum software release levels are required:

- Sun Cluster 3.1 Update 3, Update 4, or Sun Cluster 3.2 (SPARC and AMD)
- Shared Sun StorageTek QFS 4.5 with Patch 03 or Sun StorageTek QFS 4.6 (SUNWqfsr, SUNWqfsu packages)
- Solaris 9 Update 8 (SPARC and X86 clients only) or Solaris 10 (SPARC and AMD). No other operating systems are supported.

Hardware Architecture Supported

The following hardware architectures are supported:

Note – Mixed architectures are not supported.

- SPARC
- AMD64

Storage Requirements

The shared storage configuration needs to include hardware-level mirroring with RAID5 support. Servers and clients should use the Sun StorageTek Traffic Manager (MPxIO) configuration, and only shared storage is supported.

Configuration Instructions

The following examples use a configuration consisting of three SPARC Sun Cluster nodes that are identified as follows:

```
ctelab30    MDS #SPARC Sun Cluster Node
ctelab31    MDS #SPARC Sun Cluster Node
ctelab32    MDC #SPARC QFS Client Node
```

▼ To Edit the `/etc/hosts` File

- **After installation of the operating system, prepare the nodes by editing the `/etc/hosts` file on each node.**

For example:

```
### SC Cluster Nodes ###
129.152.4.57 ctelab30 # Cluster Node
129.152.4.58 ctelab31 # Cluster Node
129.152.4.59 ctelab32 # QFS Client Node

### SC Logical ###
192.168.4.100 sc-qfs1

### QFS NET ### ##
## ctelab30
192.168.4.20 ctelab30-4
192.168.4.160 ctelab30-qfe1-test
192.168.4.210 ctelab30-qfe2-test

## ctelab31
192.168.4.21 ctelab31-4
192.168.4.161 ctelab31-qfe1-test
192.168.4.211 ctelab31-qfe2-test

## ctelab32
192.168.4.22 ctelab32-qfs
```

▼ To Configure the Metadata Server Network

The following examples illustrate the setup process for the server network. These examples assume the following settings:

- `ce0, ce1` used for Sun Cluster private interconnects
- `qfe0` used for the public network (129.152.4.0)
- `qfe1, qfe2` used for IPMP group for use with the Sun StorageTek QFS private network

For this example the `/etc/hosts`, `/etc/netmasks`, `/etc/nsswitch.conf`, `/etc/hostname.qfe1`, and `/etc/hostname.qfe2` files must be modified on each server cluster node, as follows:

1. Check the `/etc/nsswitch.conf` file.

For example:

```
hosts: cluster files dns nis
```

2. Append the following to the `/etc/netmasks` file:

```
192.168.4.0 255.255.255.0
```

3. Edit the /etc/hostname.qfe1 file so that it contains the following:

```
ctelab30-4 netmask + broadcast + group qfs_ipmp1 up addif
ctelab30-qfe1-test deprecated -failover netmask + broadcast +
up
```

4. Edit the /etc/hostname.qfe2 file so that it contains the following:

```
ctelab30-qfe2-test netmask + broadcast + deprecated group
qfs_ipmp1 -failover standby up
```

▼ To Configure the Metadata Client Network

The following examples illustrate the setup process for the client network. These examples assume the following settings:

- qfe0 used for the public network (129.152.4.0)
- qfe1 used for the Sun StorageTek QFS private network. (192.168.4.0)

For this example, the /etc/hosts, /etc/netmasks, /etc/nsswitch.conf, /etc/hostname.qfe1, and /etc/hostname.qfe2 must be modified on each metadata controller (MDC) node, as follows:

1. Check the /etc/nsswitch.conf file and modify as follows:

```
hosts: files dns nis
```

2. Append the following to the /etc/netmasks file:

```
192.168.4.0 255.255.255.0
```

3. Edit the /etc/hostname.qfe1 file to contain the following:

```
ctelab32-4
```

▼ To Install and Configure Sun Cluster

After the operating system has been prepared and the nodes have the MPxIO multipathing software enabled, you can install and configure the Sun Cluster software as follows:

1. Install the Sun Cluster software, following the Sun Cluster documentation.

2. Identify shared storage devices to be used as quorum devices.

For example:

```
sccdidadm -L
scconf -a -q globaldev=dx
scconf -c -q reset
```

▼ To Configure the Sun StorageTek QFS Metadata Server

After the Sun Cluster software has been installed and the cluster configuration has been verified, you can install and configure the Sun StorageTek QFS MDS, as follows:

1. **Install the Sun StorageTek QFS software by following the instructions in the *Sun StorageTek QFS Installation and Upgrade Guide*.**

For example:

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

2. **Using the Sun Cluster command `scdidadm -L`, identify the devices that will be used for the Sun StorageTek QFS configuration.**

3. **Edit the `mcf` file to reflect the file system devices.**

For example:

```
#
# File system Qfs1 #
Qfs1 2 maQfs1onshared
/dev/did/dsk/d7s0 20 mmQfs1on
/dev/did/dsk/d8s0 21 mmQfs1on
/dev/did/dsk/d16s0 22 mrQfs1on
/dev/did/dsk/d10s0 23 mrQfs1on
/dev/did/dsk/d13s0 24 mrQfs1on
#
# File system Qfs2 #
Qfs2 5 maQfs2onshared
/dev/did/dsk/d9s0 50 mmQfs2on
/dev/did/dsk/d11s0 51 mmQfs2on
/dev/did/dsk/d17s0 52 mrQfs2on
/dev/did/dsk/d12s0 53 mrQfs2on
/dev/did/dsk/d14s0 54 mrQfs2on
/dev/did/dsk/d15s0 55 mrQfs2on
/dev/did/dsk/d18s0 56 mrQfs2on
```

4. Set local mode on the MDS Sun StorageTek QFS data devices.

For example, for the `qfs1` file system defined above, the following would be carried out for devices defined as `mr` devices:

```
#/usr/cluster/bin/scconf -r -D name=dsk/d16,nodelist=ctelab31
#/usr/cluster/bin/scconf -c -D name=dsk/d16,localonly=true
#/usr/cluster/bin/scconf -r -D name=dsk/d10,nodelist=ctelab31
#/usr/cluster/bin/scconf -c -D name=dsk/d10,localonly=true
#/usr/cluster/bin/scconf -r -D name=dsk/d13,nodelist=ctelab31
#/usr/cluster/bin/scconf -c -D name=dsk/d13,localonly=true
```

5. Edit the `/etc/opt/SUNWsamfs/defaults.conf` file.

For example:

```
trace
all = on
sam-fsd.size = 10M
sam-sharefsd.size = 10M
endtrace
```

6. Build the Sun StorageTek QFS file system hosts files.

For information on the hosts files, see the *Sun StorageTek QFS Installation and Upgrade Guide* and [“Changing the Shared Hosts File” on page 58](#).

Note – Since we communicate with the MDC outside the cluster, we need to establish Sun StorageTek QFS metadata traffic over the network. The MDC is not a member of the Sun Cluster configuration, so we use a logical host for this traffic. In this example configuration `sc-qfs1` is this hostname.

To build the shared host table on the MDS, do the following:

a. Use the Sun Cluster `scconf` command to obtain the host order information. For example:

```
# /usr/cluster/bin/scconf -p | egrep Cluster node name: |Node private
hostname:|Node ID:
```

b. Make note of the `scconf` command output. For example:

```
Cluster node name:  ctelab30
Node ID:  1
Node private hostname:  clusternode1-priv

Cluster node name:  ctelab31
Node ID:  2
Node private hostname:  clusternode2-priv
```

c. Create the shared hosts file.

For example, the `/etc/opt/SUNWsamfs/hosts.Qfs1` file would contain the following:

```
#
# MDS
# Shared MDS Host file for family set 'Qfs1'
#
#
ctelab30 clusternode1-priv,sc-qfs11-server
ctelab31 clusternode2-priv,sc-qfs12-
ctelab32 ctelab32-4--
```

d. Create the local hosts file.

For example, the `/etc/opt/SUNWsamfs/hosts.Qfs1.local` file would contain the following:

```
#
# MDS
# Local MDS Host file for family set 'Qfs1'
ctelab30      clusternode1-priv
ctelab31      clusternode2-priv
```

7. Create the file system using the `sammkfs` command.

For example:

```
# /opt/SUNWsamfs/sbin/sammkfs -S Qfs1
```

8. Prepare the mount points on each cluster node.

For example:

```
# mkdir -p /cluster/qfs1 /cluster/qfs2
```

9. Append file system entries to the `/etc/vfstab` file.

For example:

```
###
# QFS Filesystems
###
Qfs1 - /cluster/qfs1 samfs - no shared
Qfs2 - /cluster/qfs2 samfs - no shared
```

10. Mount the file systems.

For example:

```
# mount Qfs1, mount Qfs2 on each cluster node
```

11. Create the Sun Cluster MDS resource group.

Carry out the following steps to create the MDS resource group under Sun Cluster:

a. Add the QFS Resource type.

For example:

```
# /usr/cluster/bin/scrgadm -a -t SUNW.qfs
```

b. Create the MDS resource group.

For example:

```
# /usr/cluster/bin/scrgadm -a -g sc-qfs-rg -h ctelab30,ctelab31
# /usr/cluster/bin/scrgadm -c -g sc-qfs-rg -y RG_description= Metadata Server
+ MDC Clients
```

c. Add the logical hostname to the resource group.

For example:

```
# /usr/cluster/bin/scrgadm -a -L -g sc-qfs-rg -l sc-qfs1 -n
qfs_ipmp1@ctelab30,qfs_ipmp1@ctelab31
# /usr/cluster/bin/scrgadm -c -j sc-qfs1 -y RG_description= Logical Hostname
resource for sc-qfs1
```

d. Add the Sun StorageTek QFS file system resource to the MDS resource group.

For example:

```
# /usr/cluster/bin/scrgadm -a -g sc-qfs-rg -t SUNW.qfs -j fs-qfs-rs -x \
# QFSFileSystem=/cluster/qfs1,/cluster/qfs2 -y Resource_dependencies=sc-qfs1
```

e. Bring the resource group online.

For example:

```
# /usr/cluster/bin/scswitch -Z -g sc-qfs-rg
```

f. Check the status.

For example:

```
# /usr/cluster/bin/scswitch
```

▼ To Configure Sun StorageTek QFS Metadata Client

After the operating system has been installed on all metadata clients, you can proceed to Sun StorageTek QFS client installation and configuration.

Before carrying out these instructions, verify that MPxIO has been enabled and that the clients can access all disk devices.

1. Install the Sun StorageTek QFS software by following the instructions in the *Sun StorageTek QFS Installation and Upgrade Guide*.

For example:

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

2. Use the `format` command on the MDC and the Sun Cluster `scdidadm -L` command on the MDS to identify the devices that will be used for the Sun StorageTek QFS configuration.

3. Build the `mcf` files on the metadata clients.

For example:

```
#
# File system Qfs1
#
Qfs1      2  ma Qfs1 on shared
nodev    20  mm Qfs1 on
nodev    21  mm Qfs1 on
/dev/dsk/c6t600C0FF0000000000332B21D0B90000d0s0 22  mr Qfs1 on
/dev/dsk/c6t600C0FF0000000000876E9124FAF9C00d0s0 23  mr Qfs1 on
/dev/dsk/c6t600C0FF000000000004CAD7CC3CDE500d0s0 24  mr Qfs1 on
#
# File system Qfs2
# Qfs2     5  ma Qfs2 on shared
nodev    50  mm Qfs2 on
nodev    51  mm Qfs2 on
/dev/dsk/c6t600C0FF0000000000332B057D2FF100d0s0 52  mr Qfs2 on
/dev/dsk/c6t600C0FF0000000000876E975EDA6A000d0s0 53  mr Qfs2 on
/dev/dsk/c6t600C0FF0000000000876E9780ECA8100d0s0 54  mr Qfs2 on
/dev/dsk/c6t600C0FF000000000004CAD139A855500d0s0 55  mr Qfs2 on
/dev/dsk/c6t600C0FF000000000004CAD4C40941C00d0s0 56  mr Qfs2 on
```

4. Edit the `/etc/opt/SUNWsamfs/defaults.conf` file.

For example:

```
trace
all = on
sam-fsd.size = 10M
sam-sharefsd.size = 10M
endtrace
```

5. Build the Sun StorageTek QFS file system hosts files.

Use the information from the MDS hosts files and follow the examples below.

Note – For metadata communications between the MDS and the MDC, clients that are not members of the cluster must communicate over the logical host.

a. Create the shared hosts file.

For example, the `/etc/opt/SUNWsamfs/hosts.Qfs1` file would contain the following:

```
#
# MDC
# Shared Client Host file for family set 'Qfs1'
ctelab30  sc-qfs1      1      -      server
ctelab31  sc-qfs1      2      -
ctelab32  ctelab32-4    -      -
```

b. Create the local hosts file.

For example, the `/etc/opt/SUNWsamfs/hosts.Qfs1.local` file would contain the following:

```
#
# MDC
# Local Client Host file for family set 'Qfs1'
ctelab30  sc-qfs1@ctelab32-4
ctelab31  sc-qfs1@ctelab32-4
```

Note – The `/etc/opt/SUNWsamfs/hosts.QFS1.local` file is different for each client. In this example, the client is using its interface configured on `ctelab32-4` to bind to host `sc-qfs1` for metadata traffic.

6. Create the mount points on each cluster node.

For example:

```
# mkdir -p /cluster/qfs1 /cluster/qfs2
```

7. Edit the `/etc/vfstab` file.

For example:

```
###
# QFS Filesystems
###
Qfs1 - /cluster/qfs1 samfs - yes bg,shared
Qfs2 - /cluster/qfs2 samfs - yes bg,shared
```

8. Mount the file systems.

For example:

```
# mount Qfs1, mount Qfs2 on each MDC node
```

Changing the Sun StorageTek QFS Configuration

This section demonstrates how to make changes to, disable, or remove the Sun StorageTek QFS shared or unshared file system configuration in a Sun Cluster environment. It contains the following sections:

- [“To Change the Shared File System Configuration” on page 177](#)
- [“To Disable HA-NFS on a File System That Uses Raw Global Devices” on page 178](#)
- [“To Disable HA-NFS on a File System That Uses Solaris Volume Manager-Controlled Volumes” on page 179](#)
- [“To Disable HA-NFS on a Sun StorageTek QFS File System That Uses VxVM-Controlled Volumes” on page 181](#)

▼ To Change the Shared File System Configuration

This example procedure is based on the example in [“Example Configuration” on page 135](#).

1. **Log in to each node as the `oracle` user, shut down the database instance, and stop the listener:**

```
$ sqlplus "/as sysdba"
SQL > shutdown immediate
SQL > exit
$ lsnrctl stop listener
```

2. Log in to the metadata server as `superuser` and bring the metadata server resource group into the unmanaged state:

```
# scswitch -F -g qfs-rg
# scswitch -u -g qfs-rg
```

At this point, the shared file systems are unmounted on all nodes. You can now apply any changes to the file systems' configuration, mount options, and so on. You can also re-create the file systems, if necessary. To use the file systems again after re-creating them, follow the steps in ["Example Configuration" on page 135](#).

3. If you want to make changes to the metadata server resource group configuration or to the Sun StorageTek QFS software, remove the resource, the resource group, and the resource type, and verify that everything is removed.

For example, you might need to upgrade to new packages.

```
# scswitch -n -j qfs-res
# scswitch -r -j qfs-res
# scrgadm -r -g qfs-rg
# scrgadm -r -t SUNW.qfs
# scstat
```

At this point, you can re-create the resource group to define different names, node lists, and so on. You can also remove or upgrade the Sun StorageTek QFS shared software, if necessary. After the new software is installed, the metadata resource group and the resource can be re-created and can be brought online.

▼ To Disable HA-NFS on a File System That Uses Raw Global Devices

Use this general example procedure to disable HA-NFS on an unshared Sun StorageTek QFS file system that is using raw global devices. This example procedure is based on ["Example 1: HA-NFS on Raw Global Devices" on page 145](#).

1. Use the `scswitch(1M) -F -g` command to take the resource group offline:

```
# scswitch -F -g nfs-rg
```

2. Disable the NFS, Sun StorageTek QFS, and LogicalHost resource types:

```
# scswitch -n -j nfs1-res
# scswitch -n -j qfsnfs1-res
# scswitch -n -j lh-nfs1
```

3. Remove the previously configured resources:

```
# scrgadm -r -j nfs1-res
# scrgadm -r -j qfsnfs1-res
# scrgadm -r -j lh-nfs1
```

4. Remove the previously configured resource group:

```
# scrgadm -r -g nfs-rg
```

5. Clean up the NFS configuration directories:

```
# rm -fr /global/nfs
```

6. Disable the resource types used, if they were previously added and are no longer needed:

```
# scrgadm -r -t SUNW.HAStoragePlus
# scrgadm -r -t SUNW.LogicalHostname
# scrgadm -r -t SUNW.nfs
```

▼ To Disable HA-NFS on a File System That Uses Solaris Volume Manager-Controlled Volumes

Use this general example procedure to disable HA-NFS on an unshared Sun StorageTek QFS file system that is using Solstice DiskSuite/Solaris Volume Manager-controlled volumes. This example procedure is based on [“Example 2: HA-NFS on Volumes Controlled by Solstice DiskSuite/Solaris Volume Manager”](#) on page 153.

1. Take the resource group offline:

```
# scswitch -F -g nfs-rg
```

2. Disable the NFS, Sun StorageTek QFS, and LogicalHost resource types:

```
# scswitch -n -j nfs1-res
# scswitch -n -j qfsnfs1-res
# scswitch -n -j lh-nfs1
```

3. Remove the previously configured resources:

```
# scrgadm -r -j nfs1-res
# scrgadm -r -j qfsnfs1-res
# scrgadm -r -j lh-nfs1
```

4. Remove the previously configured resource group:

```
# scrgadm -r -g nfs-rg
```

5. Clean up the NFS configuration directories:

```
# rm -fr /global/nfs
```

6. Disable the resource types used, if they were previously added and are no longer needed:

```
# scrgadm -r -t SUNW.HAStoragePlus
# scrgadm -r -t SUNW.LogicalHostname
# scrgadm -r -t SUNW.nfs
```

7. Delete RAID-5 and RAID-1 sets:

```
# metaclear -s nfsdg -f d30 d20 d21 d22 d23 d11 d1 d2 d3 d4
```

8. Remove mediation detection of drive errors:

```
# metaset -s nfsdg -d -m scnode-A
# metaset -s nfsdg -d -m scnode-B
```

9. Remove the shared DID devices from the `nfsdg` disk group:

```
# metaset -s nfsdg -d -f /dev/did/dsk/d4 /dev/did/dsk/d5 \  
/dev/did/dsk/d6 /dev/did/dsk/d7
```

10. Remove the configuration of disk group `nfsdg` across nodes in the Sun Cluster system:

```
# metaset -s nfsdg -d -f -h scnode-A scnode-B
```

11. Delete the metadatabase, if it is no longer needed:

```
# metadb -d -f /dev/dsk/c0t0d0s7  
# metadb -d -f /dev/dsk/c1t0d0s7  
# metadb -d -f /dev/dsk/c2t0d0s7
```

▼ To Disable HA-NFS on a Sun StorageTek QFS File System That Uses VxVM-Controlled Volumes

Use this general example procedure to disable HA-NFS on an unshared Sun StorageTek QFS file system that is using VxVM-controlled volumes. This example procedure is based on [“Example 3: HA-NFS on VxVM Volumes” on page 160](#).

1. Take the resource group offline:

```
# scswitch -F -g nfs-rg
```

2. Disable the NFS, Sun StorageTek QFS, and `LogicalHost` resource types:

```
# scswitch -n -j nfs1-res  
# scswitch -n -j qfsnfs1-res  
# scswitch -n -j lh-nfs1
```

3. Remove the previously configured resources:

```
# scrgadm -r -j nfs1-res
# scrgadm -r -j qfsnfs1-res
# scrgadm -r -j lh-nfs1
```

4. Remove the previously configured resource group:

```
# scrgadm -r -g nfs-rg
```

5. Clean up the NFS configuration directories:

```
# rm -fr /global/nfs
```

6. Disable the resource types used, if they were previously added and are no longer needed:

```
# scrgadm -r -t SUNW.HAStoragePlus
# scrgadm -r -t SUNW.LogicalHostname
# scrgadm -r -t SUNW.nfs
```

7. Delete the subdisk:

```
# vxdg destroy nfsdg
```

8. Remove the VxVM devices:

```
# vxdisk rm fabric_0 fabric_1 fabric_2 fabric_3 fabric_4
```

High-Availability Sun StorageTek SAM Configuration Using Sun Cluster

Sun StorageTek SAM can also be configured for high availability by using Sun Cluster software. By allowing other nodes in a cluster to automatically host the archiving workload when the primary node fails, Sun Cluster software can significantly reduce downtime and increase productivity.

High-availability SAM (HA-SAM) depends on the Sun StorageTek QFS Sun Cluster agent, so this configuration must be installed with a shared Sun StorageTek QFS file system that is mounted and managed by the Sun StorageTek QFS Sun Cluster agent.

For more information see the *Sun StorageTek Storage Archive Manager Archive Configuration and Administration Guide*.

Advanced Topics

This chapter discusses advanced topics that are beyond the scope of basic system administration and usage. This chapter contains the following sections:

- [“Using Daemons, Processes, and Tracing” on page 185](#)
- [“Using the `setfa\(1\)` Command to Set File Attributes” on page 190](#)
- [“Configuring WORM-FS File Systems” on page 192](#)
- [“Accommodating Large Files” on page 207](#)
- [“Configuring a Multireader File System” on page 208](#)
- [“Using the SAN-QFS File System in a Heterogeneous Computing Environment” on page 210](#)
- [“Understanding I/O Types” on page 219](#)
- [“Increasing File Transfer Performance for Large Files” on page 220](#)
- [“Enabling Qwrite Capability” on page 223](#)
- [“Setting the Write Throttle” on page 224](#)
- [“Setting the Flush-Behind Rate” on page 225](#)
- [“Tuning the Number of Inodes and the Inode Hash Table” on page 226](#)

Using Daemons, Processes, and Tracing

It is useful to have an understanding of system daemons and processes when you are debugging. This section describes the Sun StorageTek QFS daemons and processes. It also provides information about daemon tracing.

Daemons and Processes

All Sun StorageTek QFS daemons are named in the form `sam-daemon_named`. Processes are named in a similar manner; the difference is that they do not end in the lowercase letter `d`.

TABLE 7-1 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-fsd</code>	Master daemon.
<code>sam-sharefsd</code>	Invokes the Sun StorageTek QFS shared file system daemon.
<code>sam-rpcd</code>	Controls the remote procedure call (RPC) application programming interface (API) server process.

When you run Sun StorageTek QFS software, `init` starts the `sam-fsd` daemon as part of `/etc/inittab` processing. The daemon is started at `init` levels 0, 2, 3, 4, 5, and 6. It should restart automatically in case of failure.

In a Sun StorageTek QFS 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 Sun StorageTek QFS 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.

Note – 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 StorageTek 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.

Note – Before the Sun StorageTek QFS 4U2 release, one port per file system was required. You can remove these entries from your file.

All metadata operations, block allocation and deallocation, and record locking are performed on the metadata server. The `sam-sharefsd` daemon does not keep any information. Hence, it can be stopped and restarted without causing any consistency problems for the file system.

Trace Files

Several Sun StorageTek QFS 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 Sun Microsystems 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 the `defaults.conf(4)` man page.

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 the `defaults.conf(4)` man page.

Trace File Content

Trace file messages contain the time and source of the message. The messages are produced by events in the processes. You can select the events by using directives in the `defaults.conf` file.

The default events are as follows:

- Customer notification `syslog` or `notify` file messages
- Nonfatal program errors
- Fatal `syslog` messages
- Process initiation and completion
- Other miscellaneous events

You can also trace the following events:

- Memory allocations
- Interprocess communication
- File actions

- Operator messages
- Queue contents when changed
- Other miscellaneous events

The default message elements (program name, process ID (PID), and time) are always included and cannot be excluded. Optionally, the messages can also contain the following elements:

- The date (The time is always included.)
- The source file name and line number
- The event type

Trace File Rotation

To prevent trace files from growing indefinitely, the `sam-fsd` daemon monitors the size of the trace files and periodically executes the following command:

```
/opt/SUNWsamfs/sbin/trace_rotate
```

This script moves the trace files to sequentially numbered copies. You can modify this script to suit your operation. Alternatively, you can provide this function using `cron(1)` or some other facility.

Determining Which Processes Are Being Traced

To determine which processes are being traced currently, enter the `sam-fsd(1M)` command at the command line. [CODE EXAMPLE 7-1](#) shows the output from this command.

CODE EXAMPLE 7-1 sam-fsd(1M) Command Output

```
# sam-fsd
Trace file controls:
sam-amld      /var/opt/SUNWsamfs/trace/sam-amld
              cust err fatal misc proc date
              size  0    age  0
sam-archiverd /var/opt/SUNWsamfs/trace/sam-archiverd
              cust err fatal misc proc date
              size  0    age  0
sam-catserverd /var/opt/SUNWsamfs/trace/sam-catserverd
              cust err fatal misc proc date
              size  0    age  0
sam-fsd       /var/opt/SUNWsamfs/trace/sam-fsd
              cust err fatal misc proc date
              size  0    age  0
sam-rftd      /var/opt/SUNWsamfs/trace/sam-rftd
              cust err fatal misc proc date
              size  0    age  0
sam-recycler  /var/opt/SUNWsamfs/trace/sam-recycler
              cust err fatal misc proc date
              size  0    age  0
sam-sharefsd  /var/opt/SUNWsamfs/trace/sam-sharefsd
              cust err fatal misc proc date
              size  0    age  0
sam-stagerd   /var/opt/SUNWsamfs/trace/sam-stagerd
              cust err fatal misc proc date
              size  0    age  0
sam-serverd   /var/opt/SUNWsamfs/trace/sam-serverd
              cust err fatal misc proc date
              size  0    age  0
sam-clientd   /var/opt/SUNWsamfs/trace/sam-clientd
              cust err fatal misc proc date
              size  0    age  0
sam-mgmt      /var/opt/SUNWsamfs/trace/sam-mgmt
              cust err fatal misc proc date
              size  0    age  0
```

For more information about enabling trace files, see the `defaults.conf(4)` man page and the `sam-fsd(1M)` man page.

Using the `setfa(1)` Command to Set File Attributes

Sun StorageTek 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. The following sections describe how the application programmer can use these features to select file attributes for files and directories, to preallocate file space, to specify the allocation method for the file, and to specify the disk stripe width.

For more information about implementing the features described in the following subsections, see the `setfa(1)` man page.

Selecting File Attributes for Files and Directories

The `setfa(1)` command sets attributes on a new or existing file. The file is created if it does not already exist.

You can set attributes on a directory as well as a file. When using `setfa(1)` with a directory, files and directories created within that directory inherit the attributes set in the original directory. To reset attributes on a file or directory to the default, use the `-d` (default) option. When the `-d` option is used, attributes are first reset to the default and then other attributes are processed.

Preallocating File Space

An end user can preallocate space for a file. This space is associated with a file so that no other files in the file system can use the disk addresses allocated to this file. Preallocation ensures that space is available for a given file, which avoids a file-system-full condition. Preallocation is assigned at the time of the request rather than when the data is actually written to disk.

Note that space can be wasted 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(1)` 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 the `mount_samfs(1M)` man page). 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(1)` 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 [TABLE 7-2](#).

TABLE 7-2 File Allocations and Stripe Widths

<code>-s</code> 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

Striped group devices are supported for Sun StorageTek QFS file systems only.

A user can specify that a file begin allocation on a particular striped group. If the file allocation method is round-robin, the file is allocated on the designated stripe group.

[CODE EXAMPLE 7-2](#) shows `setfa(1)` commands specifying that `file1` and `file2` be independently spread across two different striped groups.

CODE EXAMPLE 7-2 `setfa(1)` Commands to Spread Files Across Striped Groups

```
# setfa -g0 -s0 file1
# setfa -g1 -s0 file2
```

This capability is particularly important for applications that must achieve levels of performance that approach raw device speeds. For more information, see the `setfa(1)` man page.

Configuring 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. Beginning with release 4U3 of the Sun StorageTek QFS software, a WORM-FS feature became available as an add-on package called `SUNWsamfswm`. In the 4U4 software release the WORM-FS interface was modified to be compatible with the new Sun StorageTek 5310 network attached storage (NAS) appliance. The previous WORM-FS interface using `ssum` is no longer supported.

Note – The WORM-FS package (`SUNWsamfswm`) is included with the Sun StorageTek QFS software packages, but must be installed separately by using the `pkgadd` command.

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 herein simply as standard mode), which is the default, and Sun emulation compliance mode (referred to herein as emulation mode), which is designed to provide compatibility with the emulation mode of the Sun 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. There is no such restriction in emulation mode. The restriction in standard mode exists because of the nature of the retention trigger defined for NFS and FTP. For these protocols, retention is requested by specifying that the `setuid` mode be set on the file. Once a file is retained, a client will see the `setuid` mode bit set, but 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 indeed WORM-protected files.

Enabling the WORM-FS Feature

There are four mount options that can be used to enable the WORM-FS feature:

- `worm_capable`
Standard WORM mode. The WORM trigger command, `chmod 4000 file-name/directory-name`, is used to set the WORM bit on a file or directory.
- `worm_lite`
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 194](#) for more information.
- `worm_emul`
WORM emulation mode, which is designed to provide compatibility with the emulation mode of the Sun StorageTek 5320 network attached storage (NAS) appliance. This 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 Sun StorageTek 5320 network attached storage (NAS) appliance. This provides standard WORM lite functionality with a different WORM trigger. The WORM bit is set by changing a directory or file from

writable to read-only. As with the `worm_lite` option, the administrator can carry out special operations on files. See “WORM “Lite” Options” on page 194 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`.

CODE EXAMPLE 7-3 shows an example of WORM-FS mount options. The file system `samfs1` mounted at `/samfs1` is WORM-capable and has the default retention period for files set to 60 minutes.

CODE EXAMPLE 7-3 Using WORM-FS Mount Options

```
# cat /etc/vfstab
#device device mount FS fsckmountmount
#to mount to fsck pointtypepassat bootoptions
#
fd      -   /dev/fdfd-no-
/proc   -   /procproc-no-
/dev/dsk/c0t0d0s1- -swap-no-
samfs1  -   /samfs1samfs=yes worm_capable,def_retention=60
swap    -   /tmpmpfs=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 `sammkfs` will fail, unless you are using the `worm_lite` 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. To do this, create an ordinary 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.

The following are examples of using the WORM trigger for each of the four mount options using the system-wide default retention value:

Example 1. WORM trigger is chmod 4000

Simple application of the WORM trigger using standard WORM functionality:

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 -          /samfs1 samfs  -          no          bg,worm_capable

[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod 4000 WORM
[root@ns-east-44]# sls -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

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod 4000 test
[root@ns-east-44]# sls -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

[root@ns-east-44]# rm test
rm: test: override protection 444 (yes/no)? yes
rm: test not removed: Read-only file system
[root@ns-east-44]# ls
test
```

Example 2. WORM trigger is chmod 4000

Simple application of the WORM trigger using standard WORM lite functionality:

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 -          /samfs1 samfs -          no          bg,worm_lite

[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]# sls -D

WORM:
mode: drwxr-xr-x  links:  2  owner: root      group: root
length:          4096  admin id:  0  inode:   1025.1
access:          Jan 30 16:12  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

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod 4000 test
[root@ns-east-44]# sls -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

[root@ns-east-44]# rm test
rm: test: override protection 444 (yes/no)? yes
[root@ns-east-44]# ls
[root@ns-east-44]#
```

Example 3. WORM trigger is chmod -w

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

[root@ns-east-44]# mount samfs1
[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod -w WORM
[root@ns-east-44]# sls -D

WORM:
mode: drwxr-xr-x  links:  2  owner: root      group: root
length:      4096  admin id:    0  inode:    1025.1
access:      Jan 30 16:26  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

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod -w test
[root@ns-east-44]# 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

[root@ns-east-44]# rm test
rm: test: override protection 444 (yes/no)? yes
rm: test not removed: Read-only file system
[root@ns-east-44]# ls
test
```

Example 4. WORM trigger is chmod -w

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

[root@ns-east-44]# mount samfs1
[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod -w WORM
[root@ns-east-44]# 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

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod -w test
[root@ns-east-44]# 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

[root@ns-east-44]# rm test
rm: test: override protection 444 (yes/no)? yes
[root@ns-east-44]# ls
[root@ns-east-44]#
```

Note – 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`.

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 Solaris/UNIX utilities. This is due to the fact 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 202](#).

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. This is accomplished by applying the appropriate WORM trigger and allowing the file system to apply the default retention period. See [“Setting the Default Retention Period” on page 202](#) for more information.

CODE EXAMPLE 7-4 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 `sls` command to display the file’s WORM features. This example uses the default retention period of the file system (60 minutes, as set in [CODE EXAMPLE 7-3](#)).

CODE EXAMPLE 7-4 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
    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 StorageTek QFS file system:

- Normal
- Retained
- Expired

The normal state represents the state of an ordinary file in a Sun StorageTek 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 StorageTek 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. Ordinary 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 - no
bg, 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. It is important to note, although the granularity of the period is in minutes, the accuracy of the period is based on one day. Also, the function handling days, hours, and minutes does not account for leap years when determining retention periods. You must consider this when using one (or all) of these to set the default retention period.

You can also use the default retention period to set a file or directory's retention period beyond the year 2038. To do this, 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. Here is an example of using the default retention period to set a retention period on a directory and file which exceeds the year 2038.

CODE EXAMPLE 7-5 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]# sls -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]# sls -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 following section, [“Setting the Retention Period Using touch” on page 203](#). 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).

[CODE EXAMPLE 7-6](#) shows the use of the `touch` utility to set a file's retention period, followed by the application of the WORM trigger.

CODE EXAMPLE 7-6 Using `touch` and `chmod` to Set the Retention Period

```
# touch -a -t200508181125 test
# sls -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
# sls -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` is used to change the access time of the file or directory. The `-t` option specifies what time is 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` manpage for more information on these options.

To set the retention period to permanent retention, set the access time to its largest possible value: 203801182214.07.

Extending a File's Retention Period

[CODE EXAMPLE 7-7](#) shows an example of using `touch` to extend a file's retention period.

CODE EXAMPLE 7-7 Using `touch` to Extend a File's Retention Period

```
# s1s -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
# s1s -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 was 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 `s1s` to View WORM-FS Files

Use the `s1s` 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 start of the retention period is stored in the file's `modified` time field. The end of the retention period is stored in the file's `attribute` time field. This time is displayed as a calendar date. An additional line in the `sls` output shows the retention period state and duration.

[CODE EXAMPLE 7-8](#) shows an example of how `sls -D` displays a file's retention status.

CODE EXAMPLE 7-8 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 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
```

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, 2005, at 11:19 and will end on August 18, 2005, 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 the `sfind(1)` man page 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.

For example, [CODE EXAMPLE 7-9](#) shows the command to find files whose retention period expires after 12/24/2004 at 15:00.

CODE EXAMPLE 7-9 Using `sfind` to Find All WORM Files That Expire After a Certain Date

```
# 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, [CODE EXAMPLE 7-10](#) shows the command to find files for which more than 1 year, 10 days, 5 hours, and 10 minutes remain before expiration.

CODE EXAMPLE 7-10 Using `sfind` to Find All WORM Files With More Than a Specified Time Remaining

```
# 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, [CODE EXAMPLE 7-11](#) shows the command to find files that have retention periods longer than 10 days.

CODE EXAMPLE 7-11 Using `sfind` to Find All WORM Files With Longer Than a Specified Retention Period

```
# sfind -rlonger 10d
```

- `-rpermanent` – Finds files whose retention period is permanent.

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 the Sun StorageTek QFS file system, the system returns an `ENOSPC` error.
- If you are using a SAM-QFS 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 a SAM-QFS environment and your application must write a file that is larger than the disk cache, you can segment the file with the `segment(1)` command. For more information about the `segment(1)` command, see the `segment(1)` man page or see the *Sun StorageTek Storage Archive Manager Archive Configuration and Administration Guide*.

Configuring a Multireader File System

The multireader file system consists of a single writer host and multiple reader hosts. The `writer` and `reader` mount options that enable the multireader file system are compatible with Sun StorageTek QFS file systems only. The mount options are described in this section and on the `mount_samfs(1M)` man page.

You can mount the multireader file system on the single writer host by specifying the `-o writer` option with the `mount(1M)` command. The host system with the `writer` mount option is the only host system that is allowed to write to the file system. The `writer` host system updates the file system. You must ensure that only one host in a multireader file system has the file system mounted with the `writer` mount option enabled. If `-o writer` is specified, directories are written through to disk at each change and files are written through to disk at close.



Caution – The multireader file system can become corrupted if more than one writer host has the file system mounted at one time. It is the site administrator’s responsibility to 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(1M)` command. There is no limit to the number of host systems that can have the multireader file system mounted as a reader.

A major difference between the multireader file system and Sun StorageTek QFS shared file system is that the multireader host reads metadata from the disk, and the client hosts of a Sun StorageTek QFS shared file system read metadata over the network. The Sun StorageTek 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 StorageTek QFS shared file system. You can, however, specify the `reader` option.

If you want a Sun StorageTek QFS shared file system client host to be a read-only host, mount the Sun StorageTek 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 StorageTek QFS shared file system. For more information about the Sun StorageTek QFS shared file system, see [“Configuring a Sun StorageTek QFS Shared File System” on page 69](#). For more information about mount options, see the `mount_samfs(1M)` man page.

You must ensure that all readers in a multireader file system have access to the device definitions that describe the `ma` device. Copy the lines from the `mcf` file that resides on the primary metadata server host to the `mcf` files on the alternate metadata servers. After copying the lines, you might need to update the information about the disk controllers because, depending on your configuration, disk partitions might not show up the same way across all hosts.

In a multireader file system environment, the Sun StorageTek 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 StorageTek 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 changed, it is invalidated. This includes any type of access, whether through `cat(1)`, `ls(1)`, `touch(1)`, `open(2)`, or other methods. This immediate refresh rate ensures that the data is correct at the time the refresh is done, but it can affect performance. Depending on your site preferences, you can use the `mount(1M)` command’s `-o invalid=n` option to specify a refresh rate between 0 seconds and 60 seconds. If the refresh rate is set to a small value, the Sun StorageTek QFS file system reads the directory and other metadata information `n` seconds after the last refresh. More frequent refreshes result in more overhead for the system, but stale information can exist if `n` is nonzero.



Caution – If a file is open for a read on a `reader` host, there is no protection against that file’s being removed or truncated by the writer. You must use another mechanism, such as application locking, to protect the reader from inadvertent writer actions.

Using the SAN-QFS File System in a Heterogeneous Computing Environment

The SAN-QFS file system enables multiple hosts to access the data stored in a Sun StorageTek QFS system at full disk speeds. This capability can be especially useful for database, data streaming, web page services, or any application that demands high-performance, shared-disk access in a heterogeneous environment.

You can use the SAN-QFS file system in conjunction with fibre-attached devices in a storage area network (SAN). The SAN-QFS file system enables high-speed access to data through Sun StorageTek QFS software and software such as Tivoli SANergy file-sharing software. To use the SAN-QFS file system, you must have both the SANergy (2.2.4 or later) and the Sun StorageTek QFS software. For information about the levels of Sun StorageTek QFS and SANergy software that are supported, contact your Sun sales representative.

Note – In environments that include the Solaris OS and supported Linux operating systems, use the Sun StorageTek QFS shared file system, not the SAN-QFS file system, on the Solaris hosts.

For information about the Sun StorageTek QFS shared file system, see the [“Configuring a Sun StorageTek QFS Shared File System”](#) on page 69. For a comparison of the Sun StorageTek QFS shared file system and the SAN-QFS file system, see [“SAN-QFS Shared File System and Sun StorageTek QFS Shared File System Comparison”](#) on page 218.

FIGURE 7-1 depicts a SAN-QFS file system that uses both the Sun StorageTek QFS software and the SANergy software and shows that 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 Solaris OS are hosting the Sun StorageTek QFS software, and that all heterogeneous clients running an OS other than Solaris are hosting the SANergy software and the NFS software. The SAN-QFS file system’s metadata server hosts both the Sun StorageTek QFS and the SANergy software. This server acts not only as the metadata server for the file system but also as the SANergy MDC.

Note – The SANergy software is not supported on x64 hardware platforms.

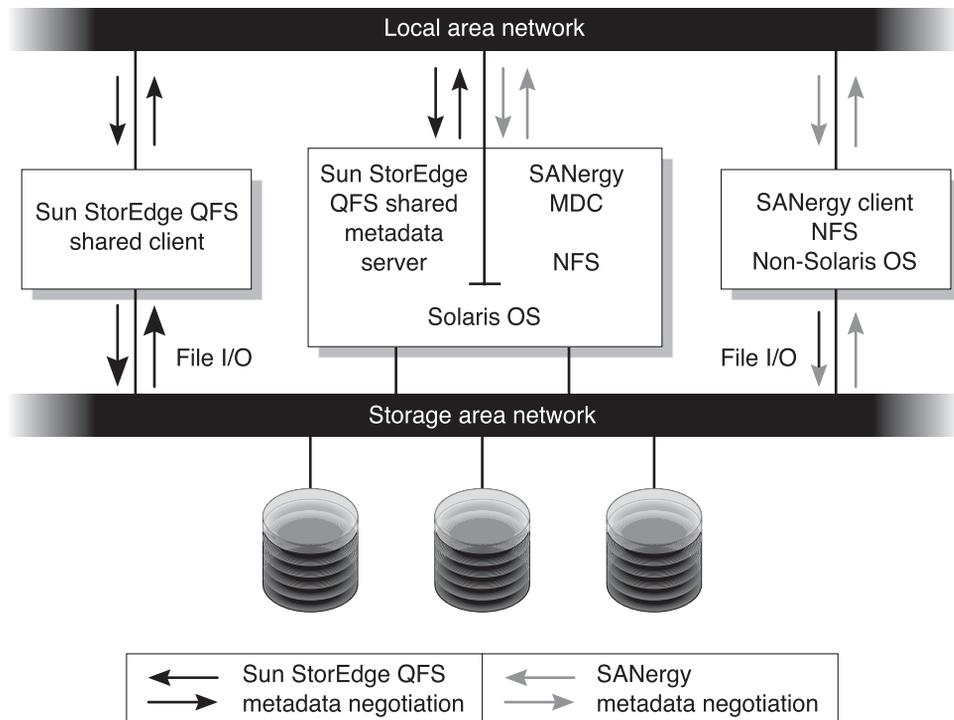


FIGURE 7-1 SAN-QFS File System Using Sun StorageTek QFS Software and SANergy Software

The rest of this section describes other aspects of the SAN-QFS file system:

- [“Before You Begin” on page 212](#)
- [“Enabling the SAN-QFS File System” on page 212](#)
- [“Unmounting the SAN-QFS File System” on page 215](#)
- [“Troubleshooting: Unmounting a SAN-QFS File System With SANergy File Holds” on page 217](#)
- [“Block Quotas in a SAN-QFS File System” on page 217](#)
- [“File Data and File Attributes in a SAN-QFS File System” on page 217](#)
- [“Using `samgrowfs\(1M\)` to Expand SAN-QFS File Systems” on page 218](#)
- [“SAN-QFS Shared File System and Sun StorageTek QFS Shared File System Comparison” on page 218](#)

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 StorageTek 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. This can result in unexpected behavior if segmented files are used within the SAN-QFS environment.
- Devices that are classified as `ms` or `md` devices in the Sun StorageTek QFS `mcf` 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.

Enabling the SAN-QFS File System

The following procedures describe how to enable the SAN-QFS file system. Perform these procedures in the order in which they are presented:

- [“To Enable the SAN-QFS File System on the Metadata Controller” on page 212](#)
- [“To Enable the SAN-QFS File System on the Clients” on page 213](#)
- [“To Install the SANergy Software on the Clients” on page 214](#)

▼ To Enable the SAN-QFS File System on the Metadata Controller

When you use the SAN-QFS file system, one host system in your environment acts as the SANergy metadata controller (MDC). This is the host system upon which the Sun StorageTek QFS file system resides.

1. **Log in to the host upon which the Sun StorageTek QFS file system resides and become `superuser`.**
2. **Verify that the Sun StorageTek QFS file system is tested and fully operational.**

3. Install and configure the SANergy software.

For instructions, see your SANergy documentation.

4. Use the `pkginfo(1)` command to verify the SANergy software release level:

```
# pkginfo -l SANergy
```

5. Ensure that the file system is mounted.

Use the `mount(1M)` command either to verify the mount or to mount the file system.

6. Use the `share(1M)` command in the following format to enable NFS access to client hosts:

```
MDC# share -F nfs -d qfs-file-system-name /mount-point
```

For *qfs-file-system-name*, specify the name of your Sun StorageTek QFS file system, such as `qfs1`. For more information about the `share(1M)` command, see the `share(1M)` or `share_nfs(1M)` man page.

For *mount-point*, specify the mount point of *qfs-file-system-name*.

7. If you are connecting to Microsoft Windows clients, configure Samba, rather than NFS, to provide security and namespace features.

To do this, add the `SANERGY_SMBPATH` environment variable in the `/etc/init.d/sanergy` file and point it to the location of the Samba configuration file. For example, if your Samba configuration file is named `/etc/swf/smb.conf`, you must add the following lines to the beginning of your `/etc/init.d/sanergy` file:

```
SANERGY_SMBPATH=/etc/sfw/smb.conf
export 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.

▼ 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 Sun sales representative.

Every client has different operational characteristics. This procedure uses general terms to describe the actions you must take to enable the SAN-QFS file system on the clients. For information specific to your clients, see the documentation provided with your client hosts.

1. **Log in to each of the client hosts.**

2. **Edit the file system defaults table on each client and add the file system.**

For example, on a Solaris OS, edit the `/etc/vfstab` file on each client and add the name of your Sun StorageTek 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 StorageTek QFS Installation and Upgrade Guide*. For information about required or suggested NFS client mount options, see your SANergy documentation.

▼ 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 StorageTek QFS file system on the MDC.

For *local-mount-point*, specify the mount point on the SANergy client.

3. Use the SANergy `fuse` command to fuse the software:

```
# fuse |mount-point
```

For *mount-point*, specify the mount point on the SANergy client.

Unmounting the SAN-QFS File System

The following procedures describe how to unmount a SAN-QFS file system that is using the SANergy software. Perform these procedures in the order in which they are presented:

- [“To Unmount the SAN-QFS File System on the SANergy Clients” on page 215](#)
- [“To Unmount the SAN-QFS File System on the Metadata Controller” on page 216](#)
- [“To Unmount the SAN-QFS File System on the Sun StorageTek QFS Clients” on page 216](#)
- [“To Unmount the SAN-QFS File System on the Sun StorageTek QFS Server” on page 216](#)

▼ To Unmount the SAN-QFS File System on the SANergy Clients

Follow these steps for each client host on which you want to unmount the SAN-QFS file system.

1. **Log in to the client host and become** `superuser`.
2. **Use the SANergy `unfuse` command to unfuse the file system from the software:**

```
# unfuse |mount-point
```

For *mount-point*, specify the mount point on the SANergy client.

3. **Use the `umount(1M)` command to unmount the file system from NFS:**

```
# umount host: /mount-point/ local-mount-point
```

For *host*, specify the MDC.

For *mount-point*, specify the mount point of the Sun StorageTek QFS file system on the MDC.

For *local-mount-point*, specify the mount point on the SANergy client.

▼ To Unmount the SAN-QFS File System on the Metadata Controller

1. **Log in to the MDC system and become superuser.**
2. **Use the `unshare(1M)` command to disable NFS access to client hosts:**

```
MDC# unshare qfs-file-system-name /mount-point
```

For *qfs-file-system-name*, specify the name of your Sun StorageTek QFS file system, such as `qfs1`. For more information about the `unshare(1M)` command, see the `unshare(1M)` man page.

For *mount-point*, specify the mount point of *qfs-file-system-name*.

▼ To Unmount the SAN-QFS File System on the Sun StorageTek QFS Clients

Follow these steps on each participating client host.

1. **Log in to a Sun StorageTek QFS client host and become superuser.**
2. **Use the `umount(1M)` command to unmount the file system.**

For example:

```
# umount /qfs1
```

▼ To Unmount the SAN-QFS File System on the Sun StorageTek QFS Server

1. **Log in to the host system upon which the Sun StorageTek QFS file system resides and become superuser.**
2. **Use the `umount(1M)` command to unmount the file system.**

Troubleshooting: Unmounting a SAN-QFS File System With SANergy File Holds

SANergy software issues holds on Sun StorageTek QFS files to reserve them temporarily for accelerated access. If SANergy crashes when holds are in effect, you will not be able to unmount the file system. If you are unable to unmount a SAN-QFS file system, examine the `/var/adm/messages` file and look for console messages that describe outstanding SANergy holds.

Whenever possible, allow the SANergy file-sharing function to clean up its holds, but in an emergency, or in case of a SANergy file-sharing system failure, use the following procedure to avoid a reboot.

▼ To Unmount a File System in the Presence of SANergy File Holds

1. Use the `unshare(1M)` command to disable NFS access.
2. Use the `samunhold(1M)` command to release the SANergy file system holds.
For more information about this command, see the `samunhold(1M)` man page.
3. Use the `umount(1M)` command to unmount the file system.

Block Quotas in a SAN-QFS File System

The SANergy software does not enforce block quotas. Therefore, it is possible for you to exceed a block quota when writing a file with the SANergy software. For more information on quotas, see [“Enabling Quotas” on page 100](#).

File Data and File Attributes in a SAN-QFS File System

The SANergy software uses the NFS software for metadata operations, which means that the NFS close-to-open consistency model is used for file data and attributes. File data and attributes among SANergy clients do not support the POSIX coherency model for open files.

Using `samgrowfs(1M)` to Expand SAN-QFS File Systems

You can use the `samgrowfs(1M)` command to increase the size of a SAN-QFS file system. To perform this task, follow the procedures described in [“Adding Disk Cache to a File System”](#) on page 64.



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(1M)` 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.

SAN-QFS Shared File System and Sun StorageTek QFS Shared File System Comparison

The SAN-QFS shared file system and the Sun StorageTek QFS shared file system have the following similarities:

- Both can stage files.
- Both are useful in data capture environments in which it is desirable that the primary file system host not be responsible for writing the data.
- Both are advantageous in environments where there is contention for writing files.

[TABLE 7-3](#) describes differences between the file systems.

TABLE 7-3 SAN-QFS Shared File System Versus Sun StorageTek QFS Shared File System

SAN-QFS File System	Sun StorageTek QFS Shared File System
Uses NFS protocol for metadata.	Uses natural metadata.
Preferred in heterogeneous computing environments (that is, when not all hosts are Sun systems).	Preferred in homogeneous Solaris OS environments.
Useful in environments where multiple, heterogeneous hosts must be able to write data.	Preferred when multiple hosts must write to the same file at the same time.

Understanding I/O Types

The Sun StorageTek QFS file systems support paged I/O, direct I/O, and switching between the I/O types. The following sections describe these I/O types.

Paged I/O

When paged I/O is used, user data is cached in virtual memory pages, and the kernel writes the data to disk. The standard Solaris OS interfaces manage paged I/O. Paged I/O (also called buffered or cached I/O) is selected by default.

Direct I/O

Direct I/O is a process by which data is transferred directly between the user's buffer and the disk. This means that much less time is spent in the system. For performance purposes, specify direct I/O only for large, block-aligned, sequential I/O.

The `setfa(1)` command and the `sam_setfa(3)` library routine both have a `-D` option that sets the direct I/O attribute for a file or directory. If applied to a directory, files and directories created in that directory inherit the direct I/O attribute. After the `-D` option is set, the file uses direct I/O.

You can also select direct I/O for a file by using the Solaris OS `directio(3C)` function call. If you use the function call to enable direct I/O, 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(1M)` command.
- Put the `forcedirectio` keyword in the mount option column of the `/etc/vfstab` file, or use it as a directive in the `samfs.cmd` file.

For more information, see the `setfa(1)`, `sam_setfa(3)`, `directio(3C)`, `samfs.cmd(4)`, and `mount_samfs(1M)` man pages.

I/O Switching

By default, paged I/O is performed, and I/O switching is disabled. However, the Sun StorageTek 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(1M)` command.

For more information about these options, see the `mount_samfs(1M)` or `samfs.cmd(4)` man pages.

Increasing File Transfer Performance for Large Files

Sun StorageTek 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.

Note – Sun recommends that you experiment with performance tuning outside of a production environment. Tuning these variables incorrectly can have unexpected effects on the overall system.

If your site has a Sun™ Enterprise Services (SES) support contract, please inform SES if you change performance tuning parameters.

▼ To Increase File Transfer Performance

1. Set the maximum device read/write directive.

The `maxphys` parameter in the Solaris `/etc/system` file controls the maximum number of bytes that a device driver reads or writes at any one time. The default value for the `maxphys` parameter can differ, depending on the level of your Sun Solaris OS, but it is typically around 128 kilobytes.

Add the following line to `/etc/system` to set `maxphys` to 8 megabytes:

```
set maxphys = 0x800000
```

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 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 StorageTek 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 `writebehind` value should be 48 kilobytes, 96 kilobytes, or some other multiple, to avoid the overhead of the read-modify-write RAID-5 parity generation.

For Sun StorageTek QFS file systems, the DAU (`sammkfs(1M) -a` command) should also be a multiple of the RAID-5 stripe size. This allocation ensures that the blocks are contiguous.

You should test the system performance after resetting the `writebehind` size. The following example shows testing timings of disk writes:

```
# timex dd if=/dev/zero of=/sam/myfile bs=256k count=2048
```

You can set the `writebehind` parameter from a mount option, from within the `samfs.cmd` file, from within the `/etc/vfstab` file, or from a command within the `samu(1M)` utility. For information about enabling this from a mount option, see the `-o writebehind=n` option on the `mount_samfs(1M)` man page. For information about enabling this from the `samfs.cmd` file, see the `samfs.cmd(4)` man page. For information about enabling this from within `samu(1M)`, see the `samu(1M)` man page.

6. Set the `readahead` parameter.

This step affects paged I/O only.

The `readahead` parameter specifies the number of bytes that are read ahead by the file system when paged I/O is being performed on a Sun StorageTek 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 is an example method of 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. It is also important to consider the

amount of memory and number of concurrent streams when you set the `readahead` value. Setting the `readahead` value multiplied by the number of streams to a value that is greater than memory can cause page thrashing.

The default `readahead` 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(1M)` utility. For information about enabling this setting from a mount option, see the `-o readahead=n` option on the `mount_samfs(1M)` man page. For information about enabling this setting from the `samfs.cmd` file, see the `samfs.cmd(4)` man page. For information about enabling this setting from within `samu(1M)`, see the `samu(1M)` man page.

7. Set the stripe width.

The `-o stripe=n` option with the `mount(1M)` command specifies the stripe width for the file system. The stripe width is based on the disk allocation unit (DAU) size. The *n* argument specifies that *n* × DAU bytes are written to one device before writing switches to the next device. The DAU size is set when the file system is initialized by the `sammkfs(1M) -a` command.

If `-o stripe=0` is set, files are allocated to file system devices using the round-robin allocation method. 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(1M)` command, see the `mount_samfs(1M)` man page. For more information about the `samfs.cmd` file, see the `samfs.cmd(4)` man page.

Enabling Qwrite Capability

By default, the Sun StorageTek QFS file systems disable simultaneous reads and writes to the same file. This is the mode defined by the UNIX `vnode` interface standard, which gives exclusive access to only one write while other writers and readers must wait. Qwrite enables simultaneous reads and writes to the same file from different threads.

The Qwrite feature can be used in database applications to enable multiple simultaneous transactions to the same file. Database applications typically manage large files and issue simultaneous reads and writes to the same file. Unfortunately, each system call to a file acquires and releases a read/write lock inside the kernel. This lock prevents overlapped (or simultaneous) operations to the same file. If the application itself implements file-locking mechanisms, the kernel-locking mechanism impedes performance by unnecessarily serializing I/O.

Qwrite can be enabled in the `/etc/vfstab` file, in the `samfs.cmd` file, and as a mount option. The `-o qwrite` option with the `mount(1M)` command bypasses the file system locking mechanisms (except for applications accessing the file system through the network file system [NFS]) and lets the application control data access. If `qwrite` is specified, the file system enables simultaneous reads and writes to the same file from different threads. This option improves I/O performance by queuing multiple requests at the drive level.

The following example uses the `mount(1M)` command to enable Qwrite on a database file system:

```
# mount -F samfs -o qwrite /db
```

For more information about this feature, see the `qwrite` directive on the `samfs.cmd(4)` man page or the `-o qwrite` option on the `mount_samfs(1M)` man page.

Setting the Write Throttle

The `-o wr_throttle=n` option limits the number of outstanding write kilobytes for one file to *n*. By default, Sun StorageTek 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 to allow the application to be resumed.

If your site has thousands of streams, such as thousands of NFS-shared workstations accessing the file system, you can tune the `-o wr_throttle=n` option in order to avoid 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 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, because this keeps more pages in memory.

Setting the Flush-Behind Rate

Two mount parameters control the flush-behind rate for pages written sequentially and 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(1M)` command.

The `flush_behind=n` mount parameter sets the maximum flush-behind value. Modified pages that are being written sequentially are written to disk asynchronously to help the 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 the `mount_samfs(1M)` man page or the `samfs.cmd(4)` man page.

Tuning the Number of Inodes and the Inode Hash Table

The Sun StorageTek QFS file system enables you to set the following two tunable parameters in the `/etc/system` file:

- `ninodes`
- `nhino`

To enable nondefault settings for these parameters, edit the `/etc/system` file, and then reboot your system.

The following subsections describe these parameters in more detail.

The `ninodes` Parameter

The `ninodes` parameter specifies the maximum number of default inodes. The value for `ninodes` determines the number of in-core inodes that Sun StorageTek 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 a Solaris tuning parameter that specifies the number of entries in the directory name look-up cache (DNLC). For more information about `ncsize`, see the *Solaris Tunable Parameters Reference Manual*.
- 2000. The file systems set `ninodes` to 2000 if the `ncsize` setting is zero or out of range.

The nhino Parameter

The `nhino` parameter specifies the size of the in-core inode hash table.

The format for this parameter in the `/etc/system` file is as follows:

```
set samfs:nhino = value
```

The range for *value* is 1 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 (2^{10}).

- 512. The file systems set `nhino` 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 (after obtaining an inode number from a directory or after extracting an inode number from an NFS file handle), a Sun StorageTek 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, it can be advantageous to set these parameter values to larger than the defaults.

If your site has file systems that contain only a small number of files, it might be advantageous to make these numbers smaller than the defaults. This could be the case, for example, if you have a file system into which you write large single-file `tar(1)` files to back up other file systems.

Troubleshooting Sun StorageTek QFS

This appendix describes some tools and procedures that can be used to troubleshoot issues with the Sun StorageTek QFS file system. Specifically, it contains the following topics:

- [“Checking File System Integrity and Repairing File Systems” on page 229](#)
- [“Troubleshooting a Failed or Hung `sammkfs\(1M\)` or `mount\(1M\)` Command in a Shared File System” on page 231](#)
- [“Troubleshooting the Linux Client” on page 246](#)

Checking File System Integrity and Repairing File Systems

Sun StorageTek QFS file systems write validation data in the following records that are critical to file system operations: directories, indirect blocks, and inodes. If the file system detects corruption while searching a directory, it issues an EDOM error, and the directory is not processed. If an indirect block is not valid, it issues an ENOCSI error, and the file is not processed. [TABLE A-1](#) summarizes these error indicators.

TABLE A-1 Error Indicators

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

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

You should monitor the following files for error conditions:

- The log file specified in `/etc/syslog.conf` for the errors shown in [TABLE A-1](#).
- The `/var/adm/messages` file for device errors.

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

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

▼ To Check a File System

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

Use this command in the following format:

```
samfsck -v family-set-name
```

For *family-set-name*, specify the name of the file system as specified in the `mcf` file.

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

- C shell:

```
# samfsck -v family-set-name |& tee file
```

- Bourne shell:

```
# samfsck -v family-set-name 2>&1 | tee file
```

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

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

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

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

▼ To Repair a File System

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

Run the `samfsck(1M)` command when the file system is not mounted. For information about unmounting a file system, see [“Unmounting a File System” on page 63](#).

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

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

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

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

Troubleshooting a Failed or Hung `sammkfs(1M)` or `mount(1M)` Command in a Shared File System

The following sections describe what to do when a `sammkfs(1M)` or `mount(1M)` command fails or when a `mount(1M)` command hangs in a shared file system.

The procedures in this section can be performed on client hosts and can also be performed on the server. Commands that can be executed only on the metadata server are preceded with a `server#` prompt.

Recovering From a Failed `sammkfs(1M)` Command

If the `sammkfs(1M)` command returns an error or messages indicating that an unexpected set of devices are to be initialized, you need to perform this procedure. It includes steps for verifying the `mcf` file and for propagating `mcf` file changes to the system.

▼ To Verify the `mcf` File and Propagate `mcf` File Changes to the System

1. Use the `sam-fsd(1M)` command to verify the `mcf` file.

For example:

```
# sam-fsd
```

Examine the output from the `sam-fsd(1M)` command and determine if there are errors that you need to fix.

2. If the output from the `sam-fsd(1M)` command indicates that there are errors in the `/etc/opt/SUNWsamfs/mcf` file, edit the `mcf` file to resolve these issues.
3. Issue the `sam-fsd(1M)` command again to verify the `mcf` file.

Repeat [Step 1](#), [Step 2](#), and [Step 3](#) of this process until the output from the `sam-fsd(1M)` command indicates that the `mcf` file is correct.

4. Issue the `samd(1M)` `config` command.

This is needed to propagate `mcf` file changes by informing the `sam-fsd` daemon of the configuration change.

For example:

```
# samd config
```

Recovering From a Failed `mount(1M)` Command

A `mount(1M)` command can fail for several reasons. This section describes some actions you can take to remedy a mount problem. If the `mount(1M)` command hangs, rather than fails, see [“Recovering From a Hung `mount\(1M\)` Command”](#) on [page 238](#).

Some failed mount(1M) behaviors and their remedies are as follows:

- If the mount(1M) command fails with a Shared server is not mounted message generated on a client, determine the server host and mount the file system on the metadata server.
- If the mount command fails with a message indicating that there is a mismatch between the file system and the mcf file, ensure the following:
 - The mcf file is syntactically valid. For more information, see [“To Verify the mcf File and Propagate mcf File Changes to the System” on page 232](#).
 - Recent changes to the mcf file are valid and have been enacted. For more information, see [“To Verify the mcf File and Propagate mcf File Changes to the System” on page 232](#).
 - The mcf file matches the server’s mcf file with device names or controller numbers adjusted for any differences on the client. You can use the samfsconfig(1M) command to diagnose some of these problems. For more information about using the samfsconfig(1M) command, see [“To Use the samfsconfig\(1M\) Command” on page 236](#).
- If the mount(1M) command fails for other reasons, use the procedures described in the following sections to verify the system characteristics that must be in place in order for the mount(1M) command to be successful. These procedures are as follows:
 - [“To Verify that the File System Can Be Mounted” on page 233](#)
 - [“To Use the samfsinfo\(1M\) and samsharefs\(1M\) Commands” on page 234](#)
 - [“To Use the samfsconfig\(1M\) Command” on page 236](#)

▼ To Verify that the File System Can Be Mounted

If this procedure does not expose errors, perform [“To Use the samfsinfo\(1M\) and samsharefs\(1M\) Commands” on page 234](#), which can help you verify that the file system has been created and that the shared hosts file is correctly initialized.

The following procedure shows you what to verify if the mount(1M) command fails.

1. Ensure that the mount point directory is present.

There are multiple ways to accomplish this. For example, you can issue the ls(1) command in the following format:

```
ls -ld mountpoint
```

For *mountpoint*, specify the name of the Sun StorageTek QFS shared file system’s mount point.

When you examine the `ls(1)` command's output, make sure that the output shows a directory with access mode 755. In other words, the codes should read `drwxr-xr-x`. [CODE EXAMPLE A-1](#) shows example output.

CODE EXAMPLE A-1 Access Mode Values

```
# ls -ld /sharefs1
drwxr-xr-x  2 root      sys           512 Mar 19 10:46 /sharefs1
```

If the access is not at this level, enter the following `chmod(1)` command:

```
# chmod 755 mountpoint
```

For *mountpoint*, specify the name of the Sun StorageTek QFS shared file system's mount point.

2. Ensure that there is an entry for the file system in the `/etc/vfstab` file.

[CODE EXAMPLE A-2](#) shows an entry for the shared file system named `sharefs1`.

CODE EXAMPLE A-2 Example `/etc/vfstab` File

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

Ensure that the `shared` flag is present in the Mount Parameters field of the shared file system's entry in the `/etc/vfstab` file.

3. Ensure that the mount point directory is not shared out for NFS use.

If the mount point is shared, use the `unshare(1M)` command to unshare it. For example:

```
# unshare mountpoint
```

For *mountpoint*, specify the name of the Sun StorageTek QFS shared file system's mount point.

▼ **To Use the `samfsinfo(1M)` and `samsharefs(1M)` Commands**

This procedure shows how to analyze the output from these commands.

1. Enter the `samfsinfo(1M)` command on the server.

Use this command in the following format:

```
samfsinfo filesystem
```

For *filesystem*, specify the name of the Sun StorageTek QFS shared file system as specified in the `mcf` file. [CODE EXAMPLE A-3](#) shows the `samfsinfo(1M)` command and output.

CODE EXAMPLE A-3 `samfsinfo(1M)` Command Example

```
titan-server# samfsinfo sharefs1
samfsinfo: filesystem sharefs1 is mounted.
name:      sharefs1      version:      2      shared
time:      Mon Apr 29 15:12:18 2002
count:     3
capacity:  10d84000      DAU:          64
space:     10180400
meta capacity: 009fe200      meta DAU:    16
meta space: 009f6c60
ord  eq   capacity      space  device
1   11   086c0000      080c39b0  /dev/dsk/c1t2100002037E9C296d0s6
2   12   086c4000      080bca50  /dev/dsk/c3t50020F2300005D22d0s6
3   13   086c4000      080a9650  /dev/dsk/c3t50020F2300006099d0s6
4   14   086c4000      08600000  /dev/dsk/c3t50020F230000651Cd0s6
```

The output from [CODE EXAMPLE A-3](#) shows a shared keyword in the following line:

```
name:      sharefs1      version:      2      shared
```

Note the list of file system devices, ordinals, and equipment numbers that appear after the following line:

```
ord  eq   capacity      space  device
```

Make sure that these numbers correspond to the devices in the file system's `mcf(4)` entry.

2. Enter the `samsharefs(1M)` command on the server.

Use this command in the following format:

```
samsharefs -R filesystem
```

For *filesystem*, specify the name of the Sun StorageTek QFS shared file system as specified in the `mcf` file. [CODE EXAMPLE A-4](#) shows the `samsharefs(1M)` command and output.

CODE EXAMPLE A-4 `samsharefs(1M)` Command Example

```
titan-server# samsharefs -R sharefs1
#
# Host file for family set `sharefs1`
#
# Version: 3      Generation: 50      Count: 4
# Server = host 0/titan, length = 216
#
titan 173.26.2.129,titan.foo.com 1 - server
tethys 173.26.2.130,tethys.foo.com 2 -
dione dione.foo.com 0 -
mimas mimas.foo.com 0 -
```

The following information pertains to the diagnostic output from the `samfsinfo(1M)` or `samsharefs(1M)` commands.

- If either command issues diagnostics or error messages, resolve them. Ensure that the output from the `samfsinfo(1M)` command includes the `shared` keyword.
- You can execute these commands on alternate server hosts and on client hosts that have no `nodev` devices in the host's `mcf(4)` entry for the file system.

If the `samfsinfo(1M)` and `samsharefs(1M)` commands do not expose irregularities, perform [“To Use the `samfsconfig\(1M\)` Command”](#) on page 236.

▼ To Use the `samfsconfig(1M)` Command

On clients with `nodev` device entries in the `mcf` file for the file system, the entire file system might not be accessible, and the shared hosts file might not be directly accessible. You can use the `samfsconfig(1M)` command to determine whether the shared file system's data partitions are accessible.

● Issue the `samfsconfig(1M)` command.

Use this command in the following format:

```
samfsconfig list-of-devices
```

For *list-of-devices*, specify the list of devices from the file system entry in the `mcf` file. Use a space to separate multiple devices in the list.

Example 1. [CODE EXAMPLE A-5](#) shows the `mcf` file for the host `tethys`, a host that does not have a `nodev` entry in its `mcf` file. It then shows the `samfsconfig(1M)` command issued.

CODE EXAMPLE A-5 `samfsconfig(1M)` Command Example Without `nodev` Entries

```
tethys# cat /etc/opt/SUNWsamfs/mcf
sharefs1          10  ma   sharefs1   on  shared
/dev/dsk/c1t210002037E9C296d0s6 11  mm   sharefs1   -
/dev/dsk/c3t50020F2300005D22d0s6 12  mr   sharefs1   -
/dev/dsk/c3t50020F2300006099d0s6 13  mr   sharefs1   -
/dev/dsk/c3t50020F230000651Cd0s6 14  mr   sharefs1   -
tethys# samfsconfig /dev/dsk/c1t210002037E9C296d0s6
/dev/dsk/c3t50020F2300005D22d0s6 /dev/dsk/c3t50020F2300006099d0s6
/dev/dsk/c3t50020F230000651Cd0s6
#
# Family Set 'sharefs1' Created Mon Apr 29 15:12:18 2002
#
sharefs1          10  ma   sharefs1   -  shared
/dev/dsk/c1t210002037E9C296d0s6 11  mm   sharefs1   -
/dev/dsk/c3t50020F2300005D22d0s6 12  mr   sharefs1   -
/dev/dsk/c3t50020F2300006099d0s6 13  mr   sharefs1   -
/dev/dsk/c3t50020F230000651Cd0s6 14  mr   sharefs1   -
```

Example 2. [CODE EXAMPLE A-6](#) shows the `samfsconfig(1M)` command being used on a host that has a `nodev` entry in its `mcf` file.

CODE EXAMPLE A-6 `samfsconfig(1M)` Command Example With `nodev` Entries

```
dione# cat /etc/opt/SUNWsamfs/mcf
sharefs1          10  ma   sharefs1   on  shared
nodev             11  mm   sharefs1   -
/dev/dsk/c4t50020F23000055A8d0s3 12  mr   sharefs1   -
/dev/dsk/c4t50020F23000055A8d0s4 13  mr   sharefs1   -
/dev/dsk/c4t50020F23000055A8d0s5 14  mr   sharefs1   -
dione# samfsconfig /dev/dsk/c4t50020F23000055A8d0s3
/dev/dsk/c4t50020F23000055A8d0s4 /dev/dsk/c4t50020F23000055A8d0s5
# Family Set 'sharefs1' Created Mon Apr 29 15:12:18 2002
# Missing slices
# Ordinal 1
# /dev/dsk/c4t50020F23000055A8d0s3 12  mr   sharefs1   -
# /dev/dsk/c4t50020F23000055A8d0s4 13  mr   sharefs1   -
# /dev/dsk/c4t50020F23000055A8d0s5 14  mr   sharefs1   -
```

For examples 1 and 2, verify that the output lists all slices from the file system, other than the metadata (mm) devices, as belonging to the file system. This is the case for example 2.

Recovering From a Hung mount(1M) Command

If the mount(1M) command hangs, follow the procedure in this section. You have a hung mount(1M) command if, for example, the mount(1M) command fails with a connection error or with a Server not responding message that does not resolve itself within 30 seconds.

The most typical remedy for a hung mount(1M) command is presented first. If that does not work, perform the subsequent procedures.

▼ To Verify Network Connections

The netstat(1M) command verifies that the sam-sharefsd daemon's network connections are correctly configured.

1. **Become superuser on the metadata server.**
2. **Type the samu(1M) command to invoke the samu(1M) operator utility.**

For example:

```
# samu
```

3. **Press :P to access the Active Services display.**

[CODE EXAMPLE A-7](#) shows a P display.

CODE EXAMPLE A-7 P Display on the Metadata Server

```
Active Services                samu  4.4 09:02:22 Sept 22 2005
Registered services for host 'titan':
    sharedfs.sharefs1
    1 service registered.
```

Examine the output. In [CODE EXAMPLE A-7](#), look for a line that contains sharedfs.*filesystem-name*. In this example, the line must contain sharedfs.sharefs1.

If no such line appears, you need to verify that both the sam-fsd and sam-sharefsd daemons have started. Perform the following steps:

- a. **Enable daemon tracing in the `defaults.conf` file.**
For information about how to enable tracing, see `defaults.conf(4)` or see [Step 2 in “To Examine the `sam-sharefsd` Trace Log” on page 244.](#)
- b. **Examine your configuration files, especially `/etc/opt/SUNWsamfs/mcf`.**
- c. **After you have checked your configuration files and verified that the daemons are active, begin this procedure again.**

4. Enter the `samsharefs(1M)` command to check the hosts file.

[CODE EXAMPLE A-11](#) shows the `samsharefs(1M)` command and correct output.

CODE EXAMPLE A-8 `samsharefs(1M) -R` Command

```
titan-server# samsharefs -R sharefs1
#
# Host file for family set `sharefs1'
#
# Version: 3      Generation: 50      Count: 4
# Server = host 0/titan, length = 216
#
titan 173.26.2.129 1 - server
tethys 173.26.2.130 2 -
dione dione 0 -
mimas mimas 0 -
```

In the output on your system, verify the following:

- The host name is present in column 1 of the output and it is designated as the server.
- The host IP address is present in column 2. If there are multiple IP addresses, make sure that they are all valid.

5. Enter the `netstat(1M)` command on the server.

[CODE EXAMPLE A-9](#) shows the `netstat(1M)` command entered on server titan.

CODE EXAMPLE A-9 `netstat(1M)` Example on the Server

```
titan-server# netstat -a | grep sam-qfs
*.sam-qfs *.*          0      0 24576  0 LISTEN
*.sam-qfs *.*          0      0 24576  0 LISTEN
titan.32834 titan.sam-qfs 32768  0 32768  0 ESTABLISHED
titan.sam-qfs titan.32891 32768  0 32768  0 ESTABLISHED
titan.sam-qfs tethys.32884 24820  0 24820  0 ESTABLISHED
titan.sam-qfs dione.35299 24820  0 24820  0 ESTABLISHED
*.sam-qfs *.*          0      0 24576  0 LISTEN
```

Verify that the output from the `netstat(1M)` command on the server contains the following:

- Three LISTEN entries.
- Two ESTABLISHED entries for the host.
- One ESTABLISHED entry for each client that is configured and running the `sam-fsd` daemon.

This example shows ESTABLISHED entries for `tethys` and `dione`. There should be one ESTABLISHED entry for each client that is configured and running, whether or not it is mounted.

6. Enter the `netstat(1M)` command on the client.

[CODE EXAMPLE A-10](#) shows the `netstat(1M)` command entered on client `dione`.

CODE EXAMPLE A-10 `netstat(1M)` Command on the Client

```
dione-client# netstat -a | grep sam-qfs
*.sam-qfs      *.*              0      0 24576      0 LISTEN
*.sam-qfs      *.*              0      0 24576      0 LISTEN
dione.32831    titan.sam-qfs   24820   0 24820      0 ESTABLISHED
*.sam-qfs      *.*              0      0 24576      0 LISTEN
```

7. Verify that the output contains the following:

- Three LISTEN entries. All entries are for the `sam-fsd` daemon.
- One ESTABLISHED entry.

If these lines are present, then the network connection is established.

If an ESTABLISHED connection is not reported, perform one or more of the following procedures:

- [“To Verify That the Client Can Reach the Server” on page 240](#)
- [“To Verify That the Server Can Reach the Client” on page 243](#)
- [“To Examine the `sam-sharefsd` Trace Log” on page 244](#)

▼ To Verify That the Client Can Reach the Server

Perform these steps if using the procedure described in [“To Verify Network Connections” on page 238](#) did not show an ESTABLISHED connection.

1. Use the `samsharefs(1M)` command to verify the hosts file on the server.

You can issue the `samsharefs(1M)` command on alternate server hosts and client hosts that have no `nodev` devices listed in the host's `mcf(4)` entry for the file system. For this step, use this command in the following format:

```
samsharefs -R filesystem
```

For *filesystem*, specify the name of the Sun StorageTek QFS shared file system as specified in the `mcf` file. [CODE EXAMPLE A-11](#) shows the `samsharefs(1M) -R` command.

CODE EXAMPLE A-11 `samsharefs(1M) -R` Command

```
titan-server# samsharefs -R sharefs1
#
# Host file for family set 'sharefs1'
#
# Version: 3      Generation: 50      Count: 4
# Server = host 0/titan, length = 216
#
titan 173.26.2.129 1 - server
tethys 173.26.2.130 2 -
dione dione 0 -
mimas mimas 0 -
```

2. Save this output.

If the steps in this procedure fail, you need this output for use in subsequent procedures.

3. Verify that the output matches expectations.

If the command fails, verify that the file system was created. In this case it is likely that one of the following has occurred:

- The `mcf` file was not created properly. You can use the `samfsconfig(1M)` command to verify the correctness of the `mcf` file.
- The file system was never created.
- The initial hosts configuration files have not been created. The configuration process involves editing the existing `mcf` file, propagating the `mcf` file changes to the rest of the system, and configuring the hosts files.

4. Find the row containing the server's name in the first column.

5. From the client, use the `ping(1M)` command on each entry from the second column of `samsharefs(1M)` output to verify that the server can be reached.

Use this command in the following format:

```
ping servername
```

For *servername*, specify the name of the server as shown in the second column of the `samsharefs(1M)` command's output.

[CODE EXAMPLE A-12](#) shows output from `ping(1M)`.

CODE EXAMPLE A-12 Using `ping(1M)` on Systems Named in `samsharefs(1M)` Output

```
dione-client# ping 173.26.2.129
ICMP Host Unreachable from gateway dione (131.116.7.218)
for icmp from dione (131.116.7.218) to 173.26.2.129
dione-client# ping titan
titan.foo.com is alive
```

6. If the `ping(1M)` command revealed unreachable hosts, examine the `hosts.filesystem.local` file from the client.

If there is more than one entry in the second column of `samsharefs(1M)` output, and if some of the entries are not reachable, ensure that only the reachable entries for the entries you want the shared file system to use are present. Also ensure that the necessary entries are present in the

`/etc/opt/SUNWsamfs/hosts.filesystem.local` file entry on that host. Ensure that the unreachable hosts are not entered in these places.

If the `sam-sharefsd` daemon attempts to connect to unreachable server interfaces, there can be substantial delays in its connecting to the server after installation, rebooting, or file system host reconfiguration. This affects metadata server failover operations substantially.

[CODE EXAMPLE A-13](#) shows the `hosts.sharefs1.local` file.

CODE EXAMPLE A-13 Examining the `hosts.filesystem.local` File

```
dione-client# cat /etc/opt/SUNWsamfs/hosts.sharefs1.local
titan          titan # no route to 173.26.2.129
tethys         tethys # no route to 173.26.2.130
```

7. If the `ping(1M)` command revealed that there were no reachable server interfaces, enable the correct server interfaces.

Either configure or initialize the server network interfaces for typical operations, or use the `samsharefs(1M)` command to update the interface names in the `hosts` file so they match the actual names.

▼ To Verify That the Server Can Reach the Client

Perform these steps if the procedure in [“To Verify Network Connections”](#) on page 238 did not show an ESTABLISHED connection.

1. Obtain `samsharefs(1M)` output.

This can be the output generated in [“To Verify That the Client Can Reach the Server”](#) on page 240, or you can generate it again using the initial steps in that procedure.

2. Find the row containing the client’s name in the first column.

3. On the client, run the `hostname(1M)` command and ensure that the output matches the name in the first column of `samsharefs(1M)` output.

[CODE EXAMPLE A-14](#) shows the `hostname(1M)` command and its output.

CODE EXAMPLE A-14 `hostname(1M)` Output

```
dione-client# hostname
dione
```

4. If the `hostname(1M)` command output matched the name in the second column of `samsharefs(1M)` output, use the `ping(1M)` command on the server to verify that the client can be reached.

[CODE EXAMPLE A-15](#) shows the `ping(1M)` command and its output.

CODE EXAMPLE A-15 `ping(1M)` Output

```
titan-server# ping dione
dione is alive
```

It is not necessary that every entry in column two of [CODE EXAMPLE A-13](#) be reachable, but all interfaces that you wish any potential server to accept connections from must be present in the column. The server rejects connections from interfaces that are not declared in the shared hosts file.

5. If the `ping(1M)` command revealed that there were no reachable client interfaces, enable the correct client interfaces.

Either configure or initialize the client network interfaces for typical operations, or use the `samsharefs(1M)` command to update the interface names in the hosts file so they match the actual names.

▼ To Examine the sam-sharefsd Trace Log

The trace log files keep information generated by the `sam-sharefsd(1M)` daemons during their operation. The trace log files include information about connections attempted, received, denied, refused, and so on, as well as other operations such as host file changes and metadata server changes.

Tracking problems in log files often involves reconciling the order of operations on different hosts by using the log files. If the hosts' clocks are synchronized, log file interpretation is greatly simplified. One of the installation steps directs you to enable the network time daemon, `xntpd(1M)`. This synchronizes the clocks of the metadata server and all client hosts during Sun StorageTek QFS shared file system operations.

The trace logs are particularly useful when setting up an initial configuration. The client logs show outgoing connection attempts. The corresponding messages in the server log files are some of the most useful tools for diagnosing network and configuration problems with the Sun StorageTek QFS shared file system. The log files contain diagnostic information for resolving most common problems.

The following procedures can resolve most `mount(1M)` problems:

- [“To Verify Network Connections” on page 238](#)
- [“To Verify That the Client Can Reach the Server” on page 240](#)
- [“To Verify That the Server Can Reach the Client” on page 243](#)

If none of the preceding procedures resolve the problem, perform the steps in this section. You can perform these steps on both the server and the client hosts.

1. Verify the presence of file `/var/opt/SUNWsamfs/trace/sam-sharefsd`.

If this file is not present, or if it shows no recent modifications, proceed to the next step.

If the file is present, use `tail(1)` or another command to examine the last few lines in the file. If it shows suspicious conditions, use one or more of the other procedures in this section to investigate the problem.

2. If Step 1 indicates that file `/var/opt/SUNWsamfs/trace/sam-sharefsd` does not exist or if the file shows no recent modifications, edit file `/etc/opt/SUNWsamfs/defaults.conf` and add lines to enable `sam-sharefsd` tracing.

- a. If a `defaults.conf` file does not already reside in `/etc/opt/SUNWsamfs`, copy the example `defaults.conf` file from `/opt/SUNWsamfs/examples/defaults.conf` to `/etc/opt/SUNWsamfs`:

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

- b. Use **vi(1)** or another editor to edit file `/etc/opt/SUNWsamfs/defaults.conf` and add lines to enable tracing.

[CODE EXAMPLE A-16](#) shows the lines to add to the `defaults.conf` file.

CODE EXAMPLE A-16 Lines to Enable Tracing in `defaults.conf`

```
trace
sam-sharefsd = on
sam-sharefsd.options = all
endtrace
```

- c. Issue the **samd(1M)** `config` command to reconfigure the **sam-fsd(1M)** daemon and cause it to recognize the new `defaults.conf` file.

For example:

```
# samd config
```

- d. Issue the **sam-fsd(1M)** `command` to check the configuration files.

[CODE EXAMPLE A-17](#) shows the output from the **sam-fsd(1M)** command.

CODE EXAMPLE A-17 Output From the **sam-fsd(1M)** Command

```
# sam-fsd
Trace file controls:
sam-archiverd off
sam-catserverd off
sam-fsd         off
sam-rftd        off
sam-recycler   off
sam-sharefsd    /var/opt/SUNWsamfs/trace/sam-sharefsd
                cust err fatal misc proc date
                size    0    age 0
sam-stagerd     off
Would stop sam-archiverd()
Would stop sam-rftd()
Would stop sam-stagealld()
Would stop sam-stagerd()
Would stop sam-initd()
```

- e. Examine the log file in `/var/opt/SUNWsamfs/trace/sam-sharefsd` to check for errors:

```
# more /var/opt/SUNWsamfs/trace/sam-sharefsd
```

3. Examine the last few dozen lines of the trace file for diagnostic information.

[CODE EXAMPLE A-18](#) shows a typical `sam-sharefsd` client log file. In this example, the server is `titan`, and the client is `dione`. This file contains normal log entries generated after a package installation, and it finishes with the daemon operating normally on a mounted file system.

CODE EXAMPLE A-18 Client Trace File

```
dione# tail -18 /var/opt/SUNWsamfs/trace/sam-sharefsd
2004-03-23 16:13:11 shf-shsam2[13835:1]: FS shsam2: Shared file system daemon
started - config only
2004-03-23 16:13:11 shf-shsam2[13835:1]: FS shsam2: Host dione
2004-03-23 16:13:11 shf-shsam2[13835:1]: FS shsam2: Filesystem isn't mounted
2004-03-23 16:13:11 shf-shsam2[13837:1]: FS shsam2: Shared file system daemon
started
2004-03-23 16:13:11 shf-shsam2[13837:1]: FS shsam2: Host dione
2004-03-23 16:13:11 shf-shsam2[13837:1]: FS shsam2: Filesystem isn't mounted
2004-03-23 16:13:11 shf-shsam2[13837:1]: FS shsam2: Kill sam-sharefsd pid 13835
2004-03-23 16:13:12 shf-shsam2[13837:1]: FS shsam2: Killed sam-sharefsd pid
13835
2004-03-23 16:13:12 shf-shsam2[13837:1]: FS shsam2: Host dione; server = titan
2004-03-23 16:13:12 shf-shsam2[13837:1]: FS shsam2: Wakened from AWAIT_WAKEUP
2004-03-23 16:13:14 shf-shsam2[13837:5]: FS shsam2: Set Client (Server titan/3).
2004-03-23 16:13:14 shf-shsam2[13837:5]: FS shsam2: SetClientSocket dione
(flags=0)
2004-03-23 16:13:14 shf-shsam2[13837:5]: FS shsam2: rdssock dione/0 (buf=6c000).
2004-03-23 16:13:15 shf-shsam2[13837:1]: FS shsam2: Signal 1 received: Hangup
2004-03-23 16:13:15 shf-shsam2[13837:1]: FS shsam2: Wakened from AWAIT_WAKEUP
2004-03-23 16:13:15 shf-shsam2[13837:1]: FS shsam2: mount; flags=18889
2004-03-23 16:18:55 shf-shsam2[13837:1]: FS shsam2: Signal 1 received: Hangup
2004-03-23 16:18:55 shf-shsam2[13837:1]: FS shsam2: Wakened from AWAIT_WAKEUP
```

Troubleshooting the Linux Client

Linux clients and Solaris clients use different procedures to locate system information and diagnose Sun StorageTek QFS issues.

Files that contain system information from the Linux kernel are in the `/proc` file system. For example the `/proc/cpuinfo` file contains hardware information.

[TABLE A-2](#) describes some files that contain useful troubleshooting information.

TABLE A-2 /proc files

File Name	Information Provided
version	Running kernel version
cpuinfo	Hardware information
uptime	Time in seconds since boot time, and total time used by processes
modules	Information about the modules that are loaded
cmdline	Command-line parameters that are passed to the kernel at boot time
filesystems	Existing file system implementations
scsi/scsi	Attached SCSI devices
fs/samfs/<QFS file system>/fsid	File system ID, which must be included in the share options for NFS

Linux kernel log messages go to the `/var/log/messages` file.

Troubleshooting Tools

Because the Linux kernel has many variations, troubleshooting problems can be very challenging. A few tools are available that might help in debugging:

- Projects such as `lkcd` and `kgdb` can provide kernel crash dump information.
- Projects such as `kdb`, `kgdb`, and `icebox` are kernel debuggers.

Note – These projects are not present by default in Red Hat Linux or SuSE. You must obtain the appropriate RPMs or SRPMs and might have to reconfigure the kernel to use them.

- The `strace` command traces system calls and signals. It is similar to the Solaris `truss` command.
- The Sun StorageTek QFS `samtrace` command dumps the trace buffer.
- The Sun StorageTek QFS `samexplorer` command generates a Sun StorageTek QFS diagnostic report script.

Note – Trace files are placed in the `/var/opt/SUNWsamfs/trace` directory on the Linux client, just as they are on the Solaris client.

Frequently Asked Questions

The following questions about the Linux client are frequently asked by users who are familiar with Sun StorageTek QFS on the Solaris platform.

- Q:** The Linux installation script reports that I got a negative score and cannot install the software. Is there any way I can still install the software?
- A:** You can try the `-force-custom` and `-force-build` installation options. However, this may cause a system panic when installing the modules. This is especially a risk if your kernel is built with some of the kernel hacking options enabled, such as spinlock debugging.
- Q:** Can I use commands such as `vmstat`, `iostat`, `top`, and `truss` on Linux?
- A:** The `vmstat`, `top`, and `iostat` commands are found in many Linux installations. If they are not installed, they can be added using the `sysstat` and `procps` RPMs. The Linux equivalents of `truss` are `ltrace` and `strace`.
- Q:** Can Sun StorageTek Traffic Manager be used with the Sun StorageTek QFS Linux client?
- A:** Yes. First build a custom kernel with multipathing support as described in the Sun StorageTek Traffic Manager documentation. Then install the Linux client software.
- Q:** Can Extensible Firmware Interface (EFI) labels be used on the Sun StorageTek QFS Linux client?
- A:** Most Linux kernels are not built with support for EFI labels with GPT (GUID Partition Table) partitions. Therefore, to use EFI labels, you must rebuild the kernel with the `CONFIG_EFI_PARTITION` option set. For more information about building a custom kernel, see the distribution documentation.
- Q:** Can I use other Linux volume managers such as logical volume management (LVM), Enterprise Volume Management System (EVMS), or Device Mapper with the Sun StorageTek QFS Linux client software?
- A:** To use a file system with EVMS, you need to have a File System Interface Module (FSIM) for that file system. No FSIM exists for the Sun StorageTek QFS product. For you to use LVM, the partition type that `fdisk` shows must be LVM(8e). Partitions that Sun StorageTek QFS uses must be SunOS.
- Q:** Can I use file systems that are larger than two terabytes?
- A:** Yes, but some utilities that provide file system information, such as `df`, might return incorrect information when run on Linux. In addition, there may be problems when sharing the file system with NFS or Samba.

- Q:** Are there any differences between the mount options supported on the Linux client and those supported on the Solaris client?
- A:** There are many `samfs` mount options that are not supported on the Linux client. Two to be aware of are `nosuid` and `forcedirectio`. See the *Sun StorageTek QFS Linux Client Guide* for a complete list of supported mount options on the Linux client.

Note – The `mdadm` (multiple devices admin) package should not be used for path failover on a Sun StorageTek QFS Linux client. The `mdadm` package writes a superblock to devices that it uses. The result is that `mdadm` has the potential to corrupt data that Solaris has written to those devices. Furthermore, Solaris can also corrupt the superblock that `mdadm` has written to the devices.

Mount Options in a Sun StorageTek QFS Shared File System

The Sun StorageTek QFS shared file system can be mounted with several mount options. This appendix describes some of these options within the context of their roles.

You can specify most mount options by using the `mount(1M)` command, by entering them in the `/etc/vfstab` file, or by entering them in the `samfs.cmd` file. For example, the following `/etc/vfstab` file includes `mount(1M)` options for a Sun StorageTek QFS shared file system:

```
sharefs1 - /sfs samfs - no shared,mh_write
```

You can change some mount options dynamically by using the `samu(1M)` operator utility. For more information about these options, see [“Using the `samu\(1M\)` Operator Utility” on page 259](#).

The following sections summarize the mount options available to you in a Sun StorageTek QFS shared file system. For more information about any of these mount options, see the `mount_samfs(1M)` man page or see the cross-references mentioned in their descriptions.

- [“Mounting in the Background: the `bg` Option” on page 252](#)
- [“Reattempting a File System Mount: the `retry` Option” on page 252](#)
- [“Declaring a Sun StorageTek QFS Shared File System: the `shared` Option” on page 253](#)
- [“Tuning Allocation Sizes: the `minallopsz=n` and `maxallopsz=n` Options” on page 253](#)
- [“Using Leases in a Sun StorageTek QFS Shared File System: the `rdlease=n`, `wrlease=n`, and `aplease=n` Options” on page 254](#)
- [“Enabling Multiple Host Reads and Writes: the `mh_write` Option” on page 255](#)

- “Setting the Minimum Number of Concurrent Threads: the `min_pool=n` Option” on page 256
- “Retaining Cached Attributes: the `meta_timeo=n` Option” on page 257
- “Specifying Striped Allocation: the `stripe` Option” on page 257
- “Specifying the Frequency With Which Metadata Is Written: the `sync_meta=n` Option” on page 258
- “Enabling WORM Functionality: the `worm_capable` and `def_retention` Options” on page 258

Mounting in the Background: the `bg` Option

The `bg` mount option specifies that if the first mount operation fails, subsequent attempts at mounting should occur in the background. By default, `bg` is not in effect, and mount attempts continue in the foreground.



Caution – Do *not* use this option if you are mounting a Sun StorageTek QFS shared file system on a Sun Cluster node or if you have shared Sun StorageTek QFS clients outside the cluster.

Reattempting a File System Mount: the `retry` Option

The `retry` mount option specifies the number of times that the system should attempt to mount a file system. The default is 10000.

Declaring a Sun StorageTek QFS Shared File System: the `shared` Option

The `shared` mount option declares a file system to be a Sun StorageTek QFS shared file system. This option must be specified in the `/etc/vfstab` file in order for the file system to be mounted as a Sun StorageTek QFS shared file system. The presence of this option in a `samfs.cmd` file or on the `mount(1M)` command does not cause an error condition, but it does not mount the file system as a Sun StorageTek QFS shared file system.

Tuning Allocation Sizes: the `minallopsz=n` and `maxallopsz=n` Options

The `-o minallopsz=n` and `-o maxallopsz=n` options to the `mount(1M)` command specify an amount of space, in kilobytes. This is the minimum block allocation size. If a file is growing, the metadata server allocates blocks when an append lease is granted. You can use the `-o minallopsz=n` option to specify the initial size of this allocation. The metadata server can increase the size of the block allocation depending on the application's access patterns up to, but not exceeding, the `-o maxallopsz=n` option's setting.

You can specify these `mount(1M)` options on the `mount(1M)` command line, in the `/etc/vfstab` file, or in the `samfs.cmd` file.

Using Leases in a Sun StorageTek QFS Shared File System: the `rdlease=n`, `wrlease=n`, and `aplease=n` Options

A *lease* grants a shared host permission to perform an operation on a file for as long as the lease is valid. The metadata server issues leases to each shared host, including itself. The leases are renewed as necessary to permit continued file operations. The possible file operations are as follows:

- A *read lease* enables existing file data to be read.
- A *write lease* enables existing file data to be overwritten.
- An *append lease* enables a file's size to be extended and enables newly allocated blocks to be written.

A shared host can continue to update leases for as long as necessary. The lease is transparent to the end user. [TABLE B-1](#) shows the mount options that enable you to specify the duration of each lease type.

TABLE B-1 Lease-Related `mount(1M)` Options

Option	Action
<code>-o rdlease=n</code>	This option specifies the maximum amount of time, in seconds, for the read lease.
<code>-o wrlease=n</code>	This option specifies the maximum amount of time, in seconds, for the write lease.
<code>-o aplease=n</code>	This option specifies the maximum amount of time, in seconds, for the append lease.

All three leases enable you to specify an n such that $15 \leq n \leq 600$. The default time for each lease is 30 seconds. A file cannot be truncated if a lease is in effect. For more information about setting these leases, see the `mount_samfs(1M)` man page.

If you change the metadata server because the current metadata server is down, you must add the lease time to the 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. For information on lease times in a Sun Cluster environment, see the *Sun StorageTek QFS Installation and Upgrade Guide*.

Enabling Multiple Host Reads and Writes: the `mh_write` Option

By default, in a Sun StorageTek QFS shared file system, multiple hosts can read the same file at the same time, and if no host is writing to that file, I/O can be paged on all hosts. Only one host can append or write to a file at any one time.

The `mh_write` option controls write access to the same file from multiple hosts. If `mh_write` is specified as a mount option on the metadata server host, the Sun StorageTek QFS shared file system enables simultaneous reads and writes to the same file from multiple hosts. If `mh_write` is not specified on the metadata server host, only one host can write to a file at any one time.

By default, `mh_write` is disabled, and only one host has write access to a file at any one time. The length of that time period is determined by the duration of the `wrlease` mount option. If the Sun StorageTek QFS shared file system is mounted on the metadata server with the `mh_write` option enabled, simultaneous reads and writes to the same file can occur from multiple hosts.

[TABLE B-2](#) describes how file access from multiple hosts is affected depending on whether the `mh_write` option is enabled on the metadata server.

TABLE B-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 and/or write. If any writer hosts exist, all I/O is direct.

The `mh_write` option does not change locking behavior. File locks behave the same whether `mh_write` is in effect or not. The `mh_write` option's effect is as follows:

- When `mh_write` is in effect, all hosts can read from and write to the same file simultaneously.
- When `mh_write` is not in effect, only one host can write to a given file during a given time interval, and no hosts can read from the file during that time interval.

Sun StorageTek QFS shared file system maintains consistency between hosts. The first time that a host executes a read or write system call, it gets a *lease*, which 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 StorageTek QFS shared file system should provide near-POSIX behavior for data reads and writes. For metadata, however, access time changes might not be seen immediately on other hosts. Changes to a file are pushed to disk at the end of a write lease, and when a read lease is acquired, the system invalidates any stale cache pages so that the newly written data can be seen.

When `mh_write` is in effect, behavior might be less consistent. When there are simultaneous readers and writers, the Sun StorageTek QFS shared file system switches all hosts accessing the file into direct I/O mode. This means that page-aligned I/O should be visible immediately to other hosts. However, non-page-aligned I/O can result in stale data being visible, or even written to the file, because the normal lease mechanism that prevents this has been disabled.

You should specify the `mh_write` option only when multiple hosts need to write to the same file simultaneously and when applications perform page-aligned I/O. In other cases, there is some risk of data inconsistency because even using `flock()` (which works with `mh_write`) to coordinate between hosts does not guarantee consistency.

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

Setting the Minimum Number of Concurrent Threads: the `min_pool=n` Option

The `min_pool=n` mount option sets the minimum number of concurrent threads for the Sun StorageTek QFS shared file system. By default, `min_pool=64` on Solaris systems, or 8 on Linux clients. This means that using default settings, there will always be at least 64 active threads in the thread pool on Solaris and 8 on Linux. You can adjust the `min_pool=n` mount option to any value between 8 and 2048, depending on the Sun StorageTek 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 on the command line.

Note – The `min_pool` mount option replaces the previous `nstreams` mount option. In version 4U6 of the software, the `nstreams` option is ignored. It will be completely removed in version 5 of the software.

Retaining Cached Attributes: the `meta_timeo=n` Option

The `meta_timeo=n` mount option determines how long the system waits between checks on the metadata information. By default, the system refreshes metadata information every three seconds. This means, for example, that an `ls(1)` command entered in a Sun StorageTek QFS shared file system with several newly created files might not return information about all the files until three seconds have passed. For n , specify a value such that $0 \leq n \leq 60$.

Specifying Striped Allocation: the `stripe` Option

By default, data files in the Sun StorageTek QFS shared file system are allocated using the round-robin file allocation method. To specify that file data be striped across disks, you can specify the `stripe` mount option on the metadata host and all potential metadata hosts. Note that by default, unshared file systems allocate file data using the striped method.

In a round-robin allocation, files are created in a round-robin fashion on each slice or striped group. This causes the maximum performance for one file to be the speed of a slice or striped group. For more information about file allocation methods, see [“Design Basics” on page 7](#).

Specifying the Frequency With Which Metadata Is Written: the `sync_meta=n` Option

You can set the `sync_meta=n` option to `sync_meta=1` or `sync_meta=0`.

By default, `sync_meta=1` and a Sun StorageTek QFS shared file system writes file metadata to disk every time the metadata changes. This slows data performance, but it ensures data consistency. This is the setting that must be in effect if you want to change the metadata server.

If you set `sync_meta=0`, the Sun StorageTek QFS shared file system writes the metadata to a buffer before writing it to disk. This delayed write delivers higher performance, but it decreases data consistency after an unscheduled machine interruption.

Enabling WORM Functionality: the `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 [“Configuring WORM-FS File Systems” on page 192](#) for more information about the WORM functionality.

Using the samu(1M) Operator Utility

This chapter shows how to use `samu(1M)` to control the devices configured within your Sun StorageTek QFS environment. Many `samu(1M)` displays are useful only for sites using the storage and archive management mechanism. If you are using `samu(1M)` in a Sun StorageTek QFS environment, some displays do not apply to you.

This chapter contains the following sections:

- [“Overview” on page 259](#)
- [“Operator Displays” on page 262](#)
- [“Operator Commands” on page 311](#)

The operations that you can perform from within `samu(1M)` can also be performed by using the `samcmd(1M)` command. For more information about `samcmd(1M)`, see the `samcmd(1M)` man page.

Overview

The `samu(1M)` operator utility requires a display terminal that displays a minimum of 24 lines by 80 characters wide. The utility includes the following features:

- Displays that enable you to monitor Sun StorageTek 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 Sun StorageTek QFS or SAM-QFS file system.

The display windows shown in this chapter are representative examples. The exact format and amount of information displayed on your terminal can be different depending on your terminal model and the devices configured in your Sun StorageTek QFS environment.

The following sections describe how to start and stop `samu(1M)`, interact with the utility, access the help windows, and view operator displays.

▼ To Invoke `samu(1M)`

- To start `samu(1M)`, type the `samu(1M)` command from the command line:

```
# samu
```

The system starts `samu(1M)` and shows the help display. This is the default initial display. To view a different `samu(1M)` display, follow the steps in [“To Display a `samu\(1M\)` Screen” on page 260](#).

The `samu(1M)` utility enables you to select its initial display. For more information about the `samu(1M)` command-line options, see the `samu(1M)` man page.

Note – `samu(1M)`, like the `vi(1)` editor, is based on the `curses(3CURSES)` routine. If you have trouble invoking `samu(1M)`, make sure that your terminal type is defined correctly.

▼ To Display a `samu(1M)` Screen

The `samu(1M)` command accepts options on its command line for displaying different `samu(1M)` screens.

1. **Type a colon (:)** to bring up the `samu(1M)` prompt.

After you type in the colon, the following appears in the lower left:

```
Command:
```

2. **Type the letter that corresponds to the display you want to view and press return.**

For example, to view the `v` display, type a `v` and press Return after the `Command:` prompt.

For a complete list of letters to type and the displays to which they correspond, see [“\(h\) - Help Display” on page 273](#).

▼ To Stop `samu(1M)`

● **To exit `samu(1M)`, type one of the following:**

- `q`
- `:q`

The `samu(1M)` operator utility exits and returns you to the command shell.

Interacting With `samu(1M)`

Interacting with `samu(1M)` is similar to interacting with the UNIX `vi(1)` editor with respect to paging forward or backward, entering commands, refreshing the display, and quitting the utility.

Each display has its own section in this chapter, and each display section shows the control key sequences you can use to navigate in that display. The `samu(1M)` man page summarizes the control key navigation sequences.

The last line of the display window shows the command and display error messages. If a command error occurs, automatic display refreshing halts until the next operator action.

Entering a Device

Each device included in the Sun StorageTek QFS environment is assigned an Equipment Ordinal (for example, 10) in the `mcf` file. Many `samu(1M)` commands reference a specific device using that Equipment Ordinal.

Example. The syntax for the `:off` command is as follows:

```
:off eq
```

For `eq`, type the Equipment Ordinal for the device you are trying to address.

Getting Online Help

When you start `samu(1M)`, the system automatically displays the first help screen.

For more information about the help (h) display, see [“\(h\) - Help Display” on page 273](#).

▼ To Access Online Help From a Display Screen

- **Type** `:h`

To move forward or backward from one screen to the next, type the following key sequence:

- Press `Ctrl-f` to page the display forward.
- Press `Ctrl-b` to page the display backward to previous pages.

You can return to the help display at any time by pressing the `h` key.

Operator Displays

You can view the `samu(1M)` operator displays by pressing the key corresponding to each display. The lowercase keys `a` through `w` display operational information.

Note – The uppercase `samu(1M)` displays (`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 chapter does not describe these uppercase displays as thoroughly as it does the lowercase displays.

For displays that overflow the screen area, the word `more` appears on the bottom of the screen display, indicating that the display contains additional information. You can use `Ctrl-f` to page forward and see more content.

[CODE EXAMPLE C-1](#) contains the word `more`, indicating that more information appears on subsequent screens.

CODE EXAMPLE C-1 samu(1M) 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
```

If `samu(1M)` prompts you to enter a device, enter its associated Equipment Ordinal. The configuration display (c) shows Equipment Ordinals for all removable media devices. To control all displays, use the control keys listed for the display.

The following sections describe the operator displays in alphabetical order. Examples are provided, and when necessary, displays are followed by a table describing the fields displayed.

Note – If you have the Sun StorageTek QFS software installed without the Sun StorageTek SAM software, only some of the operator displays will be available.

(a) - Archiver Status Display

The a display shows the archiver status.

You can invoke this display differently, depending on what you need to view, as follows:

- To display an archiver status summary, which shows the status of the archiver on a per-file-system basis, type the command with the following format:

```
Command: a
```

- To display archiving details for a specific file system, type the command with the following format:

```
Command: a filesystem
```

For *filesystem*, specify the name of a file system.

Navigation

[TABLE C-1](#) shows the control keys you can use in the `a` display.

TABLE C-1 Control Keys for the `a` Display

Key	Function
Ctrl-b	Previous file system
Ctrl-f	Next file system
Ctrl-d	Page <i>arcopies</i> forward (bottom portion)
Ctrl-u	Page <i>arcopies</i> backward (bottom portion)

[TABLE C-2](#) shows the control keys you can use in the `:a filesystem` display.

TABLE C-2 Control Keys for the `:a filesystem` Display

Key	Function
Ctrl-b	Previous file system
Ctrl-f	Next file system

Sample Display

[CODE EXAMPLE C-2](#) shows activity and statistics for a single file system in the summary display.

CODE EXAMPLE C-2 `samu(1M) a` Display

```
Archiver status                samu 4.6 07:44:02 August 8 2005
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
```

CODE EXAMPLE C-2 samu(1M) a Display (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

[TABLE C-3](#) shows the fields in the detail display.

TABLE C-3 samu(1M) a Display Field Descriptions

Field	Description
samfs1 mounted at	Mount point.
regular files	Number of regular files and their total size.
offline files	Number of offline files and their total size.
archdone files	Number of archdone files and size. Indicates that the archiver has completed processing and can perform no further processing for archdone files. Files marked as archdone have been processed for archiving but have not necessarily been archived.
copy1	Number of files and total size for archive copy 1.
copy2	Number of files and total size for archive copy 2.
copy3	Number of files and total size for archive copy 3.
copy4	Number of files and total size for archive copy 4.
Directories	Number of directories and total size.
sleeping until	Indicates when archiver runs again.

(c) - Device Configuration Display

The `c` display shows your configuration's connectivity. It lists all device names and Equipment Ordinals.

To invoke the device configuration display, type the command with the following format:

```
Command: c
```

Navigation

[TABLE C-4](#) shows the control keys you can use in this display.

TABLE C-4 Control Keys for the `c` Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-u	Half-page backward

Sample Display

[CODE EXAMPLE C-3](#) shows the device configuration display.

CODE EXAMPLE C-3 samu(1M) `c` Display

```
Device configuration:                samu      4.6 07:48:11 Sept
8 2006
ty  eq state  device_name                          fs family_set
sk  100 on    /etc/opt/SUNWsamfs/dcstkconf          100 dcL700
tp  120 off   /dev/rmt/1cbn                         100 dcL700
sg  130 on    /dev/rmt/4cbn                         100 dcL700
sg  140 on    /dev/rmt/5cbn                         100 dcL700
tp  150 off   /dev/rmt/3cbn                         100 dcL700
hy  151 on    historian                               151
```

Field Descriptions

TABLE C-5 shows the field descriptions for this display.

TABLE C-5 samu(1M) c Display Field Descriptions

Field	Description
ty	Device type.
eq	Equipment Ordinal of the device.
state	Current operating state of the device. Valid device states are as follows: <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 28.
device_name	Path to the device.
fs	Family Set Equipment Ordinal.
family_set	Name of the storage Family Set or library to which the device belongs.

(c) - Memory Display

The C display shows the content of a specified memory address. To show the content at an address, enter the address in hexadecimal.

To invoke this display, type the following command:

```
Command: C hex-address
```

For *hex-address*, specify the address of a memory location in hexadecimal. For example:

```
Command: C 0x1044a998
```

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

CODE EXAMPLE C-4 shows the memory display. The output has been truncated for inclusion in this manual.

CODE EXAMPLE C-4 samu(1M) C Display

```
Memory   base: 0x1234567                samu 4.6 07:52:25 Sept 8 2006
00000000 80921000 137ff801 edd05e21 40853420 .....x.mP^!@.4
00000010 00a00420 018528b0 01a604e0 02840640 . . .(0.&.`...@
00000020 02d030a1 a0853420 0080a0a0 100a6fff .P0! .4 . . .o.
00000030 f6921000 13c65e23 582d0000 0ba01020 v....F^#X-... .
00000040 00c45e20 48c608e0 2fd05e21 40920080 .D^ HF.`/P^!@...
00000050 037ff801 fa941000 16c45e20 48a600a0 ..x.z....D^ H&.
00000060 80921000 137ff801 d5d05e21 40853420 .....x.UP^!@.4
00000070 00a00420 018528b0 01a604e0 02840640 . . .(0.&.`...@
00000080 02d030a1 c0853420 0080a0a0 100a6fff .P0!@.4 . . .o.
00000090 f6921000 13c65e23 58a01020 00c45e20 v....F^#X . .D^
000000a0 48c608e0 2fd05e21 40920080 037ff801 HF.`/P^!@....x.
000000b0 e39405a2 00c45e20 48a600a0 80921000 c..".D^ H&. ....
000000c0 137ff801 bed05e21 40853420 00a00420 ..x.>P^!@.4 . .
000000d0 018528b0 01a604e0 02840640 02d030a1 ..(0.&.`...@.P0!
000000e0 e0853420 0080a0a0 100a6fff f6921000 `.4 . . .o.v...
000000f0 13c65e23 58a01020 00c45e20 48c608e0 .F^#X . .D^ HF.`
```

(d) - Daemon Trace Controls Display

The `d` display shows the events being traced as specified in the `defaults.conf` file. For more information about enabling trace files, see the `defaults.conf(4)` man page.

To invoke this display, type the following command:

```
Command: d
```

Sample Display

[CODE EXAMPLE C-5](#) 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.

CODE EXAMPLE C-5 samu(1M) d Display

```
Daemon trace controls          samu 4.6 07:56:38 Sept 8 2006
sam-amld      /var/opt/SUNWsamfs/trace/sam-amld
              cust err fatal misc proc debug date
              size  0    age 0
sam-archiverd /var/opt/SUNWsamfs/trace/sam-archiverd
              cust err fatal misc proc debug date
              size  0    age 0
sam-catserverd /var/opt/SUNWsamfs/trace/sam-catserverd
              cust err fatal misc proc debug date
              size  0    age 0
sam-fsd       /var/opt/SUNWsamfs/trace/sam-fsd
              cust err fatal misc proc debug date
              size  0    age 0
sam-rftd      /var/opt/SUNWsamfs/trace/sam-rftd
              cust err fatal misc proc debug date
              size  0    age 0
sam-recycler  /var/opt/SUNWsamfs/trace/sam-recycler
              cust err fatal misc proc debug date
              size  0    age 0
sam-sharefsd  /var/opt/SUNWsamfs/trace/sam-sharefsd
              cust err fatal misc proc debug date
              size  0    age 0
sam-stagerd   /var/opt/SUNWsamfs/trace/sam-stagerd
              cust err fatal misc proc debug date
              size  0    age 0
sam-serverd   /var/opt/SUNWsamfs/trace/sam-serverd
              cust err fatal misc proc debug date
              size  0    age 0
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 flags indicating the status of the VSN. These flags include *unavailable*, *read only*, and *bad media*.

To invoke this display, type the following command:

```
Command: D
```

Sample Display

[CODE EXAMPLE C-6](#) shows the device configuration display.

CODE EXAMPLE C-6 samu(1M) D Display

```
Disk volume dictionary samu      4.6 07:48:11 May 8 2007

header
version 460

volumes
magic 340322 version 9 nkeys 2 ndata 2
index  spacecapacityusedflagsvolume
   0  128824115201288778547210291200-----disk01
   1   6443827200 6443892736   70656-----disk02
clients
magic 340322 version 9 nkeys 1 ndata 1
```

Flags

[TABLE C-6](#) shows the flags for the `D` display.

TABLE C-6 Flags Field for the `samu(1M) D` Display

Field	Description
l----	Volume is labeled; <code>seqnum</code> file has been created. This is set by the administrator to prevent the software from creating a new <code>seqnum</code> file.
-r---	Volume is defined on a remote host.

TABLE C-6 Flags Field for the samu(1M) D Display (*Continued*)

Field	Description
--U--	Volume is unavailable.
---R-	Volume is read only.
----E	Media error. Set when the software detects a write error on the disk archive directory.

The `diskvols samu(1M)` command can be used to set or clear the disk volume dictionary flags. See [“The `:diskvols volume \[+flag | -flag\]` Command” on page 321](#).

(f) - File Systems Display

The `f` display shows the components of your Sun StorageTek QFS file systems.

To invoke this display, type the following command:

```
Command: f
```

Sample Display

[CODE EXAMPLE C-7](#) shows the file systems display.

CODE EXAMPLE C-7 samu(1M) f Display

```
File systems                                     samu 4.6 08:11:24 Sept 8 2006
```

ty	eq	state	device_name	status	high	low	mountpoint	server
ms	10	on	samfs1	m----	2	----	d 90% 70%	/sam1
md	11	on	/dev/dsk/c5t8d0s3					
md	12	on	/dev/dsk/c5t8d0s4					
md	13	on	/dev/dsk/c5t8d0s5					
md	14	on	/dev/dsk/c5t8d0s6					
md	15	on	/dev/dsk/c5t8d0s7					
ms	20	on	samfs2	m----	2	----	d 90% 70%	/sam2
md	21	on	/dev/dsk/c5t9d0s3					
md	22	on	/dev/dsk/c5t9d0s4					
md	23	on	/dev/dsk/c5t9d0s5					
md	24	on	/dev/dsk/c5t9d0s6					
md	25	on	/dev/dsk/c5t9d0s7					
ma	30	on	qfs1	m----	2	----	d 90% 70%	/qfs1
mm	31	on	/dev/dsk/c5t10d0s0					

CODE EXAMPLE C-7 samu(1M) f Display (Continued)

md	32	on	/dev/dsk/c5t10d0s1					
ma	40	on	qfs2	m----	2----	d	90%	70% /qfs2
mm	41	on	/dev/dsk/c5t11d0s0					
md	42	on	/dev/dsk/c5t11d0s1					
ma	50	on	qfs3	m----	2---	r-	90%	70% /qfs3
mm	51	on	/dev/dsk/c5t12d0s0					
mr	52	on	/dev/dsk/c5t12d0s1					
ma	60	on	qfs4	m----	2---	r-	90%	70% /qfs4
mm	61	on	/dev/dsk/c5t13d0s0					
mr	62	on	/dev/dsk/c5t13d0s1					
ma	100	on	shareqfs1	m----	2c--	r-	80%	70% /shareqfs1 spade
mm	101	on	/dev/dsk/c6t50020F2300004655d0s0					
mr	102	on	/dev/dsk/c6t50020F2300004655d0s1					
ma	110	on	shareqfs2	m----	2c--	r-	80%	70% /shareqfs2 spade
mm	111	on	/dev/dsk/c6t50020F2300004655d0s6					
mr	112	on	/dev/dsk/c6t50020F2300004655d0s7					

Field Descriptions

[TABLE C-7](#) shows the field descriptions for this display.

TABLE C-7 samu(1M) f Display Field Descriptions

Field	Description
ty	Device type.
eq	Equipment Ordinal of the device.
state	Current operating state of the device. Valid device states are as follows: <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 28.
device_name	File system name or path to the device.
status	Device status. For a description of status codes, see “Operator Display Status Codes” on page 306.
high	High disk usage threshold percentage.

TABLE C-7 samu(1M) f Display Field Descriptions (*Continued*)

Field	Description
low	Low disk usage threshold percentage.
mountpoint	Mount point of the file system.
server	Name of the host system upon which the file system is mounted.

(F) - Optical Disk Label Display

The F display shows the label on an optical disk.

To invoke this display, type the following command:

```
Command: F
```

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

(h) - Help Display

The h display shows a summary of the samu(1M) displays available. By default, this is the first display that the system presents when you enter the samu(1M) command at the command line.

To invoke this display, type the following command:

```
Command: h
```

Navigation

[TABLE C-8](#) shows the control keys you can use in this display.

TABLE C-8 Control Keys for the h Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Page forward (top portion)

TABLE C-8 Control Keys for the h Display (Continued)

Key	Function
Ctrl-f	Page forward
Ctrl-u	Page backward (top portion)
Ctrl-k	Toggle path display

Sample Display

There are several pages of help screens, but this manual shows only the first. Subsequent help screens show `samu(1M)` commands.

[CODE EXAMPLE C-8](#) shows the initial help screen for the Sun StorageTek SAM file system. On a Sun StorageTek QFS file system, not all of the displays appear in the initial help screen. For example, the removable media displays are not available if you are running a Sun StorageTek QFS system. If you are using the SAM-QFS configuration, the help screen appears as shown in [CODE EXAMPLE C-8](#).

CODE EXAMPLE C-8 `samu(1M)` Initial Help Screen for a Sun StorageTek SAM System

```
Help informationpage 1/15samu 4.6 08:18:13 Sept 8 2006
Displays:
  a  Archiver statuswPending stage queue
  c  Device configurationCMemory
  d  Daemon trace controlsDDisk volume dictionary
  f  File systemsFOptical disk label
  h  Help informationIInode
  l  Usage informationJPreview shared memory
  m  Mass storage statusKKernel statistics
  n  Staging statusLShared memory tables
  o  Optical disk statusMShared memory
  p  Removable media load requestsNFile system parameters
  r  Removable mediaPActive Services
  s  Device statusRSAM-Remote
  t  Tape drive statusSSector data
  u  Staging queueTSCSI sense data
  v  Robot catalogUDevice table

more (ctrl-f)
```

(I) - Inode Display

The `I` display shows the content of inodes.

You can invoke this display differently, depending on what you need to view, as follows:

- To display inodes for an entire file system, type the command with the following format:

```
Command: I filesystem
```

For *filesystem*, specify the name of a file system.

- To display a specific inode, type the command with the following format:

```
Command: I inode-number
```

For *inode-number*, specify the inode number in either hexadecimal or decimal.

Navigation

[TABLE C-9](#) shows the control keys you can use in this display.

TABLE C-9 Control Keys for the I Display

Key	Function
Ctrl-b	Previous inode
Ctrl-f	Next inode
Ctrl-k	Advance display format

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

[CODE EXAMPLE C-9](#) shows the inode display.

CODE EXAMPLE C-9 samu(1M) I Display

```
Inode      0x1 (1) format: file          samu 4.6 08:27:14 Sept 8 2006
incore: y

00008100 mode      -r-----          409cdf57 access_time
00000001 ino        (1)              1d32ea20
00000001 gen        (1)              4096b499 modify_time
00000002 parent.ino (2)              02588660
```

CODE EXAMPLE C-9 samu(1M) I Display (Continued)

```
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.

To invoke this display, type the following command:

```
Command: J
```

Navigation

TABLE C-10 shows the control keys you can use in this display.

TABLE C-10 Control Keys for the J Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-u	Half-page backward

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

[CODE EXAMPLE C-10](#) shows the preview shared memory display. This output has been truncated for inclusion in this manual.

CODE EXAMPLE C-10 samu(1M) J Display

```
Preview shared memory   size: 155648   samu 4.6 08:30:05 Sept 8 2006

00000000 00040000 00014d58 00000000 00000000  .....MX.....
00000010 00000000 00000000 73616d66 73202d20  .....samfs -
00000020 70726576 69657720 6d656d6f 72792073  preview memory s
00000030 65676d65 6e740000 00026000 00000000  egment.....
00000040 00025fff 00000000 00040000 00014d58  .._.....MX
00000050 00000000 00000000 00000000 00000000  .....
00000060 0000d9e0 00000064 00000000 000001b8  ..Y.....d.....8
00000070 3f800000 447a0000 0000d820 00000008  ?...Dz...X ....
```

(K) - Kernel Statistics Display

The K display shows kernel statistics, such as the number of inodes currently in memory.

To invoke this display, type the following command:

```
Command:K
```

Navigation

[TABLE C-11](#) shows the control keys you can use in this display.

TABLE C-11 Control Keys for the K Display

Key	Function
Ctrl-b	Page backward
Ctrl-f	Page forward

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

[CODE EXAMPLE C-11](#) shows the kernel statistics display.

CODE EXAMPLE C-11 samu(1M) K Display

```
Kernel statistics                               samu 4.6 08:33:19 Sept 8 2006

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
inofree              128577
```

(1) - 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.

To invoke this display, type the following command:

```
Command: l
```

Sample Display

[CODE EXAMPLE C-12](#) shows an example of a usage display.

CODE EXAMPLE C-12 samu(1M) l Display

```
Usage information                               samu      4.6 08:36:27 Sept 8 2006

hostid = 80e69e6e OS name: SunOS Architecture: SPARC CPUs: 2 (2 online)

library 40: capacity389.3G bytes space291.1Gbytes, usage 25%
library 51: capacity9.5G bytes space9.5Gbytes, usage 0%
library 55: capacity0bytes space0bytes, usage 0%
library 56: capacity10.7G bytes space10.7Gbytes, usage 0%
```

CODE EXAMPLE C-12 samu(1M) l Display (Continued)

```
library totals: capacity409.5G bytes space311.3Gbytes, usage 24%

filesystem samfs3: capacity54.5Mbytes space13.4Mbytes, usage 75%
filesystem samfs4: capacity319.5Mbytes space298.0Mbytes, usage 7%
filesystem samfs7: capacity96.6Mbytes space69.6Mbytes, usage 28%
filesystem samfs6: capacity5.0Gbytes space4.9Gbytes, usage 3%
filesystem samfs8: capacity5.0Gbytes space4.9Gbytes, usage 2%
filesystem totals: capacity10.5Gbytes space10.2Gbytes, usage 3%
```

Note – In versions of the software before 4U3, 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.

To invoke this display, type the following command:

```
Command: L
```

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

[CODE EXAMPLE C-13](#) shows the shared memory tables.

CODE EXAMPLE C-13 samu(1M) L Display

```
Shared memory tables                               samu 4.6 08:38:31 May  8 2007

shm ptr tbl:                                       defaults:
size          12000 (73728)                        optical      mo
left          44c8 (17608)                         tape         lt
scanner pid   1861                                  timeout      600
fifo path     01b0 /var/opt/SUNWsamfs/previews      stages       100
dev_table     01cc                                  log_facility 184
first_dev     0450                                  dio minfilesize 100
scan_mess     cf50
```

CODE EXAMPLE C-13 samu(1M) L Display (Continued)

preview_shmid	1	label barcode	FALSE
flags	0x20000000	barcodes low	FALSE
preview stages	55776	export unavail	FALSE
preview avail	100	attended	TRUE
preview count	0	start rpc	FALSE
preview sequence	445		
age factor	1	vsn factor	1000
fs tbl ptr	0xd820	fs count	8
fseq 10 samfs1 state	0 0 0	0 0	
fseq 20 samfs2 state	0 0 0	0 0	
fseq 30 qfs1 state	0 0 0	0 0	
fseq 40 qfs2 state	0 0 0	0 0	
fseq 50 qfs3 state	0 0 0	0 0	
fseq 60 qfs4 state	0 0 0	0 0	
fseq 100 shareqfs1 state	0 0 0	0 0	0
fseq 110 shareqfs2 state	0 0 0	0 0	0

(m) - Mass Storage Status Display

The m display shows the status of mass storage file systems and their member drives. This display shows only mounted file systems.

To invoke this display, type the following command:

```
Command:m
```

Sample Display

[CODE EXAMPLE C-14](#) shows the m display. Member drives are indented one space and appear directly below the file system to which they belong.

CODE EXAMPLE C-14 samu(1M) m Display

Mass storage status										samu 4.6 08:41:11 Sept 8 2006					
ty	eq	status	use	state	ord	capacity	free	ra	part	high	low				
ms	10	m----2----	1%	on		68.354G	68.343G	1M	16	90%	70%				
	md	11		1%	on	0	13.669G	13.666G							
	md	12		1%	on	1	13.669G	13.667G							
	md	13		1%	on	2	13.669G	13.667G							
	md	14		1%	on	3	13.674G	13.672G							
	md	15		1%	on	4	13.674G	13.672G							
ms	20	m----2----	1%	on		68.354G	68.344G	1M	16	90%	70%				

CODE EXAMPLE C-14 samu(1M) m Display (Continued)

md	21		1% on	0	13.669G	13.667G			
md	22		1% on	1	13.669G	13.667G			
md	23		1% on	2	13.669G	13.667G			
md	24		1% on	3	13.674G	13.672G			
md	25		1% on	4	13.674G	13.672G			
ma	30	m----2----d	4% on		64.351G	61.917G	1M	16	90% 70%
mm	31		1% on	0	4.003G	3.988G	[8363840 inodes]		
md	32		4% on	1	64.351G	61.917G			
ma	40	m----2----d	1% on		64.351G	64.333G	1M	16	90% 70%
mm	41		1% on	0	4.003G	3.997G	[8382784 inodes]		
md	42		1% on	1	64.351G	64.333G			
ma	50	m----2---r-	1% on		64.351G	64.333G	1M	16	90% 70%
mm	51		1% on	0	4.003G	3.997G	[8382784 inodes]		
mr	52		1% on	1	64.351G	64.333G			
ma	60	m----2---r-	1% on		64.351G	64.331G	1M	16	90% 70%
mm	61		1% on	0	4.003G	3.997G	[8382784 inodes]		
mr	62		1% on	1	64.351G	64.331G			
ma	100	m----2c--r-	2% on		270.672G	265.105G	1M	16	80% 70%
mm	101		1% on	0	2.000G	1.988G	[4168992 inodes]		
mr	102		2% on	1	270.672G	265.469G			
ma	110	m----2c--r-	3% on		270.656G	263.382G	1M	16	80% 70%
mm	111		1% on	0	2.000G	1.987G	[4167616 inodes]		
mr	112		2% on	1	270.656G	264.736G			

Field Descriptions

TABLE C-12 shows the field descriptions for this display.

TABLE C-12 samu(1M) m Display Field Descriptions

Field	Description
ty	Device type.
eq	Equipment Ordinal of the mass storage device.
status	Device status. For a description of status codes, see “Operator Display Status Codes” on page 306.
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	Readahead size in kilobytes.

TABLE C-12 samu(1M) m Display Field Descriptions (*Continued*)

Field	Description
part	Partial stage size in kilobytes.
high	High disk usage threshold percentage.
low	Low disk usage threshold percentage.

(M) - Shared Memory Display

The M display shows the raw shared memory segment in hexadecimal. This is a device table.

To invoke this display, type the following command:

```
Command: M
```

Navigation

[TABLE C-13](#) shows the control keys you can use in this display.

TABLE C-13 Control Keys for the M Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-u	Half-page backward

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

[CODE EXAMPLE C-15](#) shows the shared memory display. The output has been truncated for inclusion in this manual.

CODE EXAMPLE C-15 samu(1M) M Display

```
Shared memory          size: 73728          samu 4.6 08:43:20 May  8 2007

00000000 00040000 00014d58 00000000 00000000 .....MX.....
00000010 00000000 00000000 73616d66 73202d20 .....samfs -
00000020 73686172 6564206d 656d6f72 79207365 shared memory se
00000030 676d656e 74000000 00012000 000044c8 gment..... ..DH
00000040 0000dd20 00000000 00000742 00000745 ..] .....B...E
00000050 00000001 00000000 00000000 c0000000 .....@...
00000060 00000001 0001534d 00000000 00000000 .....SM.....
00000070 00000000 00000000 00000000 00000000 .....

00000080 00000000 00000000 00000000 00000000 .....
00000090 20000000 000001b0 000001cc 00000450 .....0...L...P
000000a0 0000cf50 00000001 00000001 4c696365 ..OP.....Lice
000000b0 6e73653a 204c6963 656e7365 206e6576 nse: License nev
000000c0 65722065 78706972 65732e00 00000000 er expires.....
000000d0 00000000 00000000 00000000 00000000 .....
000000e0 00000000 00000000 00000000 00000000 .....
000000f0 00000000 00000000 00000000 00000000 .....
```

(n) - Staging Status Display

The `n` display shows the status of the stager for all media. It displays a list of outstanding stage requests.

You can invoke this display differently, depending on what you need to view, as follows:

- To display the staging status for all staging activity, type the command with the following format:

```
Command:n
```

- To display the staging status for a specific media type, type the command with the following format:

```
Command:n mt
```

For *mt*, specify one of the media types shown in the *mc f(4)* man page.

Sample Display

[CODE EXAMPLE C-16](#) shows the staging status display.

CODE EXAMPLE C-16 samu(1M) n Display

```
Staging status                               samu 4.6 08:47:16 May  8 2007

Log output to: /var/opt/SUNWsamfs/stager/log

Stage request: dt.DAT001
Loading VSN DAT001

Staging queues
ty pid  user      status  wait files vsn
dt 16097 root      active  0:00   12 DAT001
```

(N) - File System Parameters Display

The *N* display shows all mount point parameters, the superblock version, and other file system information.

To invoke this display, type the following command:

```
Command: N
```

Navigation

[TABLE C-14](#) shows the control keys you can use in this display.

TABLE C-14 Control Keys for the *N* Display

Key	Function
Ctrl-b	Previous file system
Ctrl-d	Page partitions forward

TABLE C-14 Control Keys for the N Display (Continued)

Key	Function
Ctrl-f	Next file system
Ctrl-i	Detailed status interpretations
Ctrl-u	Page partitions backward

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

[CODE EXAMPLE C-17](#) shows the file system parameters display.

CODE EXAMPLE C-17 samu(1M) N Display

```
File system parameters                               samu 4.6 08:55:19 Sept 8 2006

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 Disk Status Display

The `o` display shows the status of all optical disk drives configured within the environment.

To invoke this display, type the following command:

```
Command: o
```

Navigation

[TABLE C-15](#) shows the control keys you can use in this display.

TABLE C-15 Control Keys for the `o` Display

Key	Function
Ctrl-b	Page backward
Ctrl-f	Page forward
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-k	Select (manual, automated library, both, priority)
Ctrl-u	Half-page backward

Sample Display

[CODE EXAMPLE C-18](#) shows the optical disk status display.

CODE EXAMPLE C-18 samu(1M) `o` Display

```
Optical disk status          samu    4.6 Thu Oct 11 13:15:40
ty  eq  status      act  use  state  vsn
mo 35  --1---wo-r    1 29%  ready  oper2
```

Field Descriptions

TABLE C-16 shows the field descriptions for this display.

TABLE C-16 samu(1M) o Display Field Descriptions

Field	Description
ty	Device type.
eq	Equipment Ordinal of the optical disk.
status	Device status. For a description of status codes, see “Operator Display Status Codes” on page 306.
act	Activity count.
use	Percentage of cartridge space used.
state	Current operating state of the optical disk. Valid device states are as follows: <ul style="list-style-type: none">• <code>ready</code>—The device is on, and the disk is loaded in the transport; available for access.• <code>notrdy</code>—The device is on, but no disk 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.
vsn	Volume serial name assigned to the optical disk, or the keyword <code>noLabel</code> if the volume is not labeled.

(p) - Removable Media Load Requests Display

The `p` display lists information about pending load requests for removable media. You can use the `mt` argument to select either a specific type of media, such as DLT tape, or a family of media, such as tape. The priority display lists the priority in the preview queue, rather than the user queue, and sorts the entries by priority.

It displays mount requests in the following formats:

- Both manual and automated library requests by user
- Both manual and automated library requests by priority
- Manual requests only
- Automated library requests only

You can invoke this display differently, depending on what you need to view, as follows:

- To display mount requests for all removable devices currently selected, type the command with the following format:

```
Command: p
```

- To display mount requests for devices of a given removable media type, type the command with the following format:

```
Command: p mt
```

For *mt*, specify one of the media types shown in the `mcf(4)` man page.

Navigation

[TABLE C-17](#) shows the control keys you can use in this display.

TABLE C-17 Control Keys for the `p` Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-k	Toggle between the different display formats
Ctrl-u	Half-page backward

Sample Display

[CODE EXAMPLE C-19](#) shows the removable media load requests display.

CODE EXAMPLE C-19 `samu(1M) p` Display

```
Removable media load requests all both samu 4.6 09:14:19 Sept 8 2006
count: 1

index type pid      user      rb  flags      wait count  vsn
  0  dt  15533  root      150 W--f---   0:00      DAT001
```

Field Descriptions

[TABLE C-18](#) shows the field descriptions for this display.

TABLE C-18 samu(1M) p Display Field Descriptions

Field	Description
index	Index number in the preview table.
type	Device type code assigned to the removable media.
pid	UNIX process identifier. A process identifier of 1 indicates NFS access.
user	Name assigned to the user requesting the load.
priority	Priority of the request.
rb	Equipment Ordinal of the automated library in which the requested VSN resides.
flags	Flags for the device. See TABLE C-19 .
wait	The elapsed time since the mount request was received.
count	The number of requests for this VSN, if it is a stage.
vsn	Volume serial name of the volume.

Flags

[TABLE C-19](#) shows the flags for the p display.

TABLE C-19 Flags Field for the samu(1M) p Display

Field	Description
w-----	Write access requested.
-b-----	Entry is busy.
--C----	Clear VSN requested.
---f---	File system requested.
----N--	Media is foreign to the 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 StorageTek QFS single port multiplexer.

To invoke this display, type the following command:

```
Command: P
```

Navigation

[TABLE C-20](#) shows the control keys you can use in this display.

TABLE C-20 Control Keys for the P Display

Key	Function
Ctrl-b	Page backward
Ctrl-f	Page forward

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

[CODE EXAMPLE C-20](#) shows the active services display.

CODE EXAMPLE C-20 samu(1M) P Display

```
Active Services                               samu      4.6 09:08:33 Sept 8 2006

Registered services for host 'pup':
  sharedfs.qfs2
  sharedfs.qfs1
  2 services registered.
```

(r) - Removable Media Status Display

The r display enables you to monitor the activity on removable media devices such as tape drives. You can monitor either a specific type of device, such as video tape, or a family of devices such as all tape devices.

You can invoke this display differently, depending on what you need to view, as follows:

- To display the status for all removable media devices, type the command with the following format:

```
Command: r
```

- To display the status for a specific device, type the command with the following format:

```
Command: r eq
```

For *eq*, specify the Equipment Ordinal for the device.

Sample Display

[CODE EXAMPLE C-21](#) shows the removable media status display.

CODE EXAMPLE C-21 samu(1M) r Display

```
Removable media status: all                samu 4.6 09:11:27 Sept 8 2006
ty  eq  status      act  use  state  vsn
dt 150  --1-----r   0  63%  ready  DAT001
```

Field Descriptions

[TABLE C-21](#) shows the field descriptions for this display.

TABLE C-21 samu(1M) r Display Field Descriptions

Field	Description
ty	Device type.
eq	Equipment Ordinal of the drive.
status	Device status. For a description of status codes, see “Operator Display Status Codes” on page 306 .
act	Activity count.
use	Percentage of cartridge space used.

TABLE C-21 samu(1M) r Display Field Descriptions (*Continued*)

Field	Description
state	Current operating state of the removable media. Valid device states are as follows: <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>notrdy</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 28.
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.

To invoke this display, type the following command:

```
Command: R
```

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

(s) - Device Status Display

The `s` display shows the status for all devices configured within the environment.

To invoke this display, type the following command:

```
Command: s
```

Navigation

TABLE C-22 shows the control keys you can use in this display.

TABLE C-22 Control Keys for the s Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-u	Half-page backward

Sample Display

CODE EXAMPLE C-22 shows the device status display.

CODE EXAMPLE C-22 samu(1M) s Display

```
Device status                               samu      4.6 09:14:05 Sept 8 2006

ty      eq state  device_name                fs status  pos
sk      100 on    /etc/opt/SUNWsamfs/dcstkconf 100 m-----r
        stk_dismount(2275) 0, volser 700073
sg      120 on    /dev/rmt/2cbn                100 -----p
        empty
sg      130 on    /dev/rmt/5cbn                100 --l----o-r
        Ready for data transfer
sg      140 on    /dev/rmt/6cbn                100 -----p
        empty
sg      150 on    /dev/rmt/4cbn                100 -----p
        empty
hy      151 on    historian                      151 -----
```

Field Descriptions

TABLE C-23 shows the field descriptions for this display.

TABLE C-23 samu(1M) s Display Field Descriptions

Field	Description
ty	Device type.
eq	Equipment ordinal 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 Ordinal of the family, set to which the device belongs.
status	Device status. For a description of status codes, see “Operator Display Status Codes” on page 306 .

(S) - Sector Data Display

The S display shows raw device data.

To invoke this display, type the following command:

```
Command: S
```

Navigation

TABLE C-24 shows the control keys you can use in this display.

TABLE C-24 Control Keys for the S Display

Key	Function
Ctrl-b	Previous sector
Ctrl-d	Page forward (top portion)
Ctrl-f	Next sector
Ctrl-k	Advance display format
Ctrl-u	Page backward (top portion)

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

(t) - Tape Drive Status Display

The t display shows the status of all tape drives configured within the environment.

To invoke this display, type the following command:

```
Command: t
```

Navigation

[TABLE C-25](#) shows the control keys you can use in this display.

TABLE C-25 Control Keys for the t Display

Key	Function
Ctrl-b	Page backward
Ctrl-f	Page forward

Sample Display

[CODE EXAMPLE C-23](#) shows the tape drive status display.

CODE EXAMPLE C-23 samu(1M) t Display

```
Tape drive status                                samu      4.6 09:21:07 Sept 8 2006
ty  eq  status      act  use  state  vsn
sg 120  -----p    0   0%  notrdy
      empty
sg 130  -----p    0   0%  notrdy
      empty
sg 140  -----p    0   0%  notrdy
      empty
sg 150  --l-----r    0  41%  ready  700088
      idle
```

Field Descriptions

TABLE C-26 shows the field descriptions for this display.

TABLE C-26 samu(1M) t Display Field Descriptions

Field	Description
ty	Device type.
eq	Equipment Ordinal of the drive.
status	Device status. For a description of status codes, see “Operator Display Status Codes” on page 306 .
act	Activity count.
use	Percentage of cartridge space used.
state	Current operating state of the removable media. Valid device states are as follows: <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>notrdy</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 28.
vsn	Volume serial name assigned to the volume, or the keyword <code>nolabel</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.

To invoke this display, type the following command:

```
Command: T
```

Navigation

[TABLE C-27](#) shows the control keys you can use in this display.

TABLE C-27 Control Keys for the T Display

Key	Function
Ctrl-b	Previous equipment
Ctrl-f	Next equipment

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

(u) - Staging Queue Display

The u display lists all files in the staging queue.

To invoke this display, type the following command:

```
Command:u
```

Navigation

[TABLE C-28](#) shows the control keys you can use in this display.

TABLE C-28 Control Keys for the u Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-k	Display the path on the second line of each entry
Ctrl-u	Half-page backward

Sample Display

CODE EXAMPLE C-24 shows the staging queue display.

CODE EXAMPLE C-24 samu(1M) u Display

```
Staging queue by media type: all          samu 4.6 09:24:23 Sept 8 2006
volumes 1 files 22
```

ty	length	fseq	ino	position	offset	vsu
dt	451.611k	20	1030	207cc	473	DAT001
dt	341.676k	20	1031	207cc	7fc	DAT001
dt	419.861k	20	1032	207cc	aa9	DAT001
dt	384.760k	20	1033	207cc	df2	DAT001
dt	263.475k	20	1034	207cc	10f5	DAT001
dt	452.901k	20	1035	207cc	1305	DAT001
dt	404.598k	20	1036	207cc	1690	DAT001
dt	292.454k	20	1037	207cc	19bb	DAT001
dt	257.835k	20	1038	207cc	1c05	DAT001
dt	399.882k	20	1040	207cc	1e0b	DAT001
dt	399.882k	40	1029	208d7	2	DAT001
dt	257.835k	40	1030	208d7	323	DAT001
dt	292.454k	40	1031	208d7	528	DAT001
dt	404.598k	40	1032	208d7	772	DAT001
dt	452.901k	40	1033	208d7	a9d	DAT001
dt	263.475k	40	1034	208d7	e28	DAT001
dt	384.760k	40	1035	208d7	1038	DAT001
dt	419.861k	40	1036	208d7	133b	DAT001
dt	341.676k	40	1037	208d7	1684	DAT001
dt	451.611k	40	1038	208d7	1931	DAT001
dt	161.326k	40	1039	208d7	1cba	DAT001
dt	406.400k	40	1040	208d7	1dfe	DAT001

Field Descriptions

TABLE C-29 shows the field descriptions for this display.

TABLE C-29 samu(1M) u Display Field Descriptions

Field	Description
ty	Device type.
length	File length.
fseq	File system equipment number.
ino	The inode number.

TABLE C-29 samu(1M) u Display Field Descriptions (*Continued*)

Field	Description
position	The position of the archive file on the specific medium.
offset	Offset of the archive file on the specific medium.
vsn	Volume serial name of the volume.

(U) - Device Table Display

The U display shows the device table in a human-readable form.

You can invoke this display differently, depending on what you need to view, as follows:

- To display the device table for all devices, type the command with the following format:

```
Command:U
```

- To display the device table for a specific device, type the command with the following format:

```
Command:U eq
```

For *eq*, specify the Equipment Ordinal of the device.

Navigation

[TABLE C-30](#) shows the control keys you can use in this display.

TABLE C-30 Control Keys for the U Display

Key	Function
Ctrl-b	Previous equipment
Ctrl-f	Next equipment

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

[CODE EXAMPLE C-25](#) shows the device table display.

CODE EXAMPLE C-25 samu(1M) U Display

```
Device table: eq: 10      addr: 00000450  samu 4.6 09:28:40 Sept 8 2006

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 cataloged in the automated library.

You can invoke this display differently, depending on what you need to view, as follows:

- To display the catalog for all devices, type the command with the following format:

```
Command: v
```

- To display catalog information for a specific device, type the command with the following format:

```
Command: v eq
```

For *eq*, specify the Equipment Ordinal of the device. Type the keyword *historian* to view the historian catalog.

At certain times, *samu(1M)* prompts for a device to be entered, as follows:

```
Enter robot: eq
```

For *eq*, specify the Equipment Ordinal of the device or press return. Pressing return displays information for the previous device specified.

For a list of all device names and Equipment Ordinals, see [“\(c\) - Device Configuration Display” on page 265](#).

Navigation

[TABLE C-31](#) shows the control keys you can use in this display.

TABLE C-31 Control Keys for the v 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 enter one of the following to select a sort key: <ul style="list-style-type: none"> 1 - sort by slot. 2 - sort by count.

TABLE C-31 Control Keys for the v Display (Continued)

Key	Function
	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

[CODE EXAMPLE C-26](#) shows the automated library catalog display.

CODE EXAMPLE C-26 samu(1M) v Display

```

Robot VSN catalog by slot      : eq 100samu      4.6 09:30:25 Sept 8 2006
count 32
slot      access time count use flags      ty vsn
0      2004/05/08 08:35   64   0% -il-o-b----- sg 700071
1      2004/05/08 09:08   27  12% -il-o-b----- sg 700073
2      2004/05/08 09:12   26  12% -il-o-b----- sg 700077
3      2004/05/08 08:39   37  40% -il-o-b----- sg 700079
4      2004/05/08 09:16   24   6% -il-o-b----- sg 700084
5      2004/05/08 09:18   24  41% -il-o-b----- sg 700088
6      none                0   0% -il-o-b----- sg 700090
7      none                0   0% -il-o-b----- sg 700092
8      none                0   0% -il-o-b----- sg 000155
9      none                0   0% -il-o-b----- sg 000156
10     none                0   0% -il-o-b----- sg 000157
11     none                0   0% -il-o-b----- sg 000158
12     none                0   0% -il-o-b----- sg 000154
13     none                0   0% -il-o-b----- sg 000153
14     none                0   0% -il-o-b----- sg 000152

```

Field Descriptions

[TABLE C-32](#) shows the field descriptions for this display.

TABLE C-32 samu(1M) v Display Field Descriptions

Field	Description
Robot VSN catalog	Name of the specified automated library and time the display refreshed.
count	Number of slots allocated in this library's catalog.
slot	Slot number within the specified library.
access time	Time the volume was last accessed.
count	Number of accesses to this volume since the last audit operation.
use	Percentage of space used for the volume.
flags	Flags for the device. See TABLE C-33 for information about the flags.
ty	Device type.
vsn	Volume serial name of the volume.

Flags

[TABLE C-33](#) shows the flags from the `flags` field in [TABLE C-32](#). In some cases, more than one flag can occur in a field, and one flag overrides the other.

TABLE C-33 Flags Field for samu(1M) v Display

Flags	Description
A-----	Volume needs audit.
-i-----	Slot in use.
--l-----	Labeled. Overrides N.
--N-----	Unlabeled. This volume is foreign to the 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.

TABLE C-33 Flags Field for `samu(1M) v Display` (Continued)

Flags	Description
-----w----	Write protect. Set when the physical write protection mechanism is enabled on a cartridge.
-----R---	Read only.
-----c--	Recycle.
-----d-	Duplicate VSN. Overrides <code>U</code> .
-----U-	Volume unavailable.
-----f	Archiver found volume full.
-----X	Export slot.

(w) - Pending Stage Queue

The `w` display shows queued stage requests for which the volumes have not yet been loaded.

You can invoke this display differently, depending on what you need to view, as follows:

- To display the pending stage queue for all media, type the command with the following format:

```
Command: w
```

- To display the pending stage queue for a specific media type, type the command with the following format:

```
Command: w mt
```

For `mt`, specify one of the media types shown in the `mc f(4)` man page.

Navigation

TABLE C-34 shows the control keys you can use in this display.

TABLE C-34 Control Keys for the w Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-k	Display the path on the second line of each entry
Ctrl-u	Half-page backward

Sample Display

CODE EXAMPLE C-27 shows the pending stage queue.

CODE EXAMPLE C-27 samu(1M) w Display

```
Pending stage queue by media type: all      samu      4.6 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

TABLE C-35 shows the field descriptions for this display.

TABLE C-35 samu(1M) w Display Field Descriptions

Field	Description
ty	Device type.
length	File length.
fseq	File system Equipment Ordinal.
ino	The inode number.
position	The position (in decimal format) of the archive file on the specific medium.
offset	Offset of the archive file on the specific medium.
vsn	Volume serial name of the volume.

Operator Display Status Codes

The operator displays have different status codes for the removable media device displays and the file system displays. The following sections describe these 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).

The status codes in this section do not apply to the *samu(1M)* *f*, *m*, and *v* displays. For information about the status codes for the *f* and *m* displays, see [“File System Display Status Codes” on page 307](#). For information about the status codes for the *v* display, see [“\(v\) - Automated Library Catalog Display” on page 300](#).

TABLE C-36 defines the valid status codes for each position.

TABLE C-36 Removable Media Device Display Status Codes

Status Bit	Meaning for a Device
s-----	Media is being scanned.
m-----	The automated library is operational.
M-----	Maintenance mode.

TABLE C-36 Removable Media Device Display Status Codes (*Continued*)

Status Bit	Meaning for a Device
-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.
-----R	Device is ready and the media is read-only.
-----r	Device is spun up and ready.
-----p	Device is present.
-----W	Device is write protected.

File System Display Status Codes

The `f` and `m` operator displays show status codes for file systems. Status codes are displayed in an 11-position format, reading from left (position 1) to right (position 11).

The status codes in this section do not apply to the `samu(1M)` `c`, `o`, `r`, `s`, `t`, or `v` displays. For information about the status codes for the `o`, `r`, `s`, and `t` displays, see [“Removable Media Device Display Status Codes” on page 306](#). For information about the status codes for the `v` display, see [“\(v\) - Automated Library Catalog Display” on page 300](#).

TABLE C-37 defines the valid status codes for each position.

TABLE C-37 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 StorageTek SAM file system version 1.
-----2-----	Sun StorageTek SAM file system version 2.
-----c----	Sun StorageTek QFS shared file system.
-----W---	Single writer.
-----R--	Multireader.
-----r-	mr devices.
-----d	md devices.

Operator Display Device States

The *c*, *m*, *o*, *r*, *s*, and *t* operator displays show device state codes. These codes represent the current access state for the device. TABLE C-38 defines the valid state codes.

TABLE C-38 Operator Display Device States

Device State	Description
on	The device is available for access. For certain displays, this state might be superseded by the states <i>ready</i> or <i>notrdy</i> .
ro	The device is available for read-only access. For certain displays, this state might be superseded by the states <i>ready</i> or <i>notrdy</i> .

TABLE C-38 Operator Display Device States (*Continued*)

Device State	Description
<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 <code>NOT READY</code> 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>notrdy</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(1M)</code> and <code>unload(1M)</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 28 .

You can use the `samu(1M)` `down`, `off`, and `on` device state commands to change device states to `down`, `off`, or `on`. You can enter these commands from any `samu(1M)` display, but if you enter them from the `c`, `m`, `o`, `r`, `s`, or `t` display, you can see the device state change in the display. For example, you could set a device state to `off` from within the `P` display, but you would not be able to see the new device state reflected in the display.

The following procedures show what to type to change a device’s state from `down` to `on` and from `on` to `down`.

▼ To Change a Drive State from `down` to `on`

1. **Bring up a `samu(1M)` display that shows drive and automated library device states.**

The following `samu(1M)` displays all show device states: `c`, `m`, `o`, `r`, `s`, and `t`.

2. **Visually inspect the display to verify that the device is in the down state.**

3. **Type** :off.

Turning the device off halts all activity so the device can be started cleanly in the next step. For example:

```
Command:off eq
```

For *eq*, specify the Equipment Ordinal of the device.

4. **Type** :on.

For example:

```
Command:on eq
```

For *eq*, specify the Equipment Ordinal of the device.

▼ To Change a Drive State from on to down

1. **Bring up a samu(1M) display that shows drive and automated library device states.**

The following samu(1M) displays all show device states: c, m, o, r, s, and t.

2. **Visually inspect the display to verify that the device is in the on state.**

3. **Type** :off.

Turning the device off halts all activity so the device can be stopped cleanly in the next step. For example:

```
Command:off eq
```

For *eq*, specify the Equipment Ordinal of the device.

4. **Type** :down.

For example:

```
Command:down eq
```

For *eq*, specify the Equipment Ordinal of the device.

Operator Commands

The following sections describe the operator commands that you can enter from the `samu(1M)` operator utility's command interface. You can enter the commands from any display.

The following types of operator commands are available:

- “Device Commands” on page 311
- “File System Commands: I/O Management” on page 312
- “File System Commands: Direct I/O Management” on page 314
- “File System Commands: Sun StorageTek QFS Shared File Systems” on page 316
- “File System Commands: Miscellaneous” on page 317
- “Miscellaneous Commands” on page 321

If you want to enter any operator commands from the Solaris OS command line, you must use them as arguments to the `samcmd(1M)` command. For more information about the `samcmd(1M)` command, see the `samcmd(1M)` man page.

In the following subsections, each `samu(1M)` command is prefaced with a colon (:) when it is entered to designate that a command is being entered and not a series of hot keys.

Device Commands

TABLE C-39 shows the device commands and their actions.

TABLE C-39 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 StorageTek SAM software to attempt to use this drive.

TABLE C-39 Device Command Actions (*Continued*)

Command	Action
<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 future allocation to this device. For more information, see “Per-logical unit number (LUN) Allocation Control” on page 28.
<code>alloc</code>	Removes the <code>nalloc</code> flag from the device. The <code>nalloc</code> flag prohibits any future 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 28.

All of these commands are used in the following format: `:command eq`. For *eq*, specify the Equipment Ordinal of the device.

File System Commands: I/O Management

The following commands enable you to manage I/O characteristics dynamically.

The `:flush_behind eq value` Command

The `flush_behind` command sets the maximum `flush_behind value`. When set to a value greater than 0, modified pages that are being written sequentially are written to disk asynchronously to help the Solaris kernel layer keep the pages clean. This option sets the maximum `flush_behind` value.

For *value*, specify an integer number of kilobytes such that $0 \leq \text{value} \leq 8192$. By default, *value*=0, which disables `flush_behind`.

For *eq*, specify the Equipment Ordinal for the file system.

The `:force_nfs_async eq` and `:noforce_nfs_async eq` Commands

These commands 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 `noforce_nfs_async` command, which is the default, synchronously writes data through to disk.

The `force_nfs_async` command is effective only if the file system is mounted as an NFS server and only if the clients are mounted with the `noac` NFS mount option. For more information about mounting an NFS file system, see the `mount_nfs(1M)` man page.

For *eq*, specify the Equipment Ordinal for the file system.



Caution – The `force_nfs_async` option violates NFS protocols. Use this command with caution. In the event of a server interruption, data can be lost. Data is cached on the NFS server and cannot be seen immediately by all the clients if there are multiple NFS servers. Multiple NFS servers can be enabled within the Sun StorageTek QFS shared file system. For more information about the Sun StorageTek QFS shared file system, see [“Configuring a Sun StorageTek QFS Shared File System” on page 69](#).

The `:readahead eq contig` Command

The `readahead` command specifies the maximum number of bytes that can be read ahead by the file system.

For *eq*, specify the Equipment Ordinal for the file system.

For *contig*, specify units of 1-kilobyte blocks. This must be an integer such that $1 < contig < 8192$. The *contig* specified is truncated to a multiple of 8 kilobytes. The default *contig* is 8 (131072 bytes).

For example, the following command sets the maximum contiguous block size to 262,144 bytes for the file system defined as Equipment Ordinal 3:

```
:readahead 3 256
```

This value can also be configured in the `samfs.cmd` file by specifying the `readahead` directive. For more information, see the `samfs.cmd(4)` man page.

The `:sw_raid eq` and `:nosw_raid eq` Commands

These options specify whether or not the file system aligns the writebehind buffer. Specify `sw_raid` if the software RAID feature of a package such as Solstice DiskSuite™ is also used on this file system. The default setting is `nosw_raid`.

For *eq*, specify the Equipment Ordinal for a file system.

The `:writebehind eq contig` Command

The `writebehind` command specifies the maximum number of bytes that can be written behind by a file system.

For `eq`, specify the Equipment Ordinal for a file system.

For `contig`, specify units of 1-kilobyte blocks. This must be an integer such that $1 < contig < 8192$. The default `contig` is 8 (131072 bytes).

For example, the following command sets the maximum contiguous block size to 262,144 bytes for the file system defined as Equipment Ordinal 50:

```
:writebehind 50 256
```

This value can also be configured in the `samfs.cmd` file by specifying the `writebehind` directive. For more information, see the `samfs.cmd(4)` man page.

The `:wr_throttle eq value` Command

The `wr_throttle` command sets the number of outstanding write bytes for one file to `value` kilobytes.

For `eq`, specify the Equipment Ordinal for a file system.

For `value`, specify an integer number of kilobytes. If `value=0`, there is no limit. The default is 16384.

File System Commands: Direct I/O Management

The commands in this section control I/O on Sun StorageTek QFS file systems. They enable you to change the type of I/O for an individual file based on I/O size and history. If direct I/O is specified for a file, for example, through the `setfa(1)` command, these options are ignored and all I/O to regular files is direct, if possible.

These commands refer to both well-aligned and misaligned I/O. *Well-aligned* I/O occurs when the file offset falls on a 512-byte boundary and when the length of the I/O transfer is at least 512 bytes. *Misaligned* I/O occurs when the file offset does not fall on a 512-byte boundary and the length of the transfer is less than 512 bytes.

For more information about I/O and I/O management, see [“Advanced Topics” on page 185](#).

The `:dio_rd_form_min eq value` and `:dio_wr_form_min eq value` Commands

These commands set the lower limits for well-aligned I/O to *value* 1024-byte blocks. Use the `dio_rd_form_min` command to set the *value* for reads, and use the `dio_wr_form_min` command to set the *value* for writes.

For *eq*, specify the Equipment Ordinal for the file system.

For *value*, specify an integer number of 1024-byte blocks to use for the lower limit. By default, *value*=256. If *value*=0, automatic I/O switching is disabled.

The `:dio_rd_ill_min eq value` and `:dio_wr_ill_min eq value` Commands

These commands set the lower limit for misaligned I/O to *value* 1024-byte blocks. Use the `dio_rd_ill_min` command to set the *value* for reads, and use the `dio_wr_ill_min` command to set the *value* for writes.

For *eq*, specify the Equipment Ordinal for the file system.

For *value*, specify an integer number of 1024-byte blocks to use for the lower limit. By default, *value*=256. If *value*=0, automatic I/O switching is disabled.

The `:dio_rd_consec eq value` and `:dio_wr_consec eq value` Commands

These commands set the number of consecutive I/O transfers that can occur, with a buffer size greater than the specified lower limits, to *value* operations.

For *eq*, specify the Equipment Ordinal for the file system.

For *value*, specify the number of consecutive I/O transfers with a buffer size greater than the specified lower limit. The specified lower limit is the *value* of `dio_rd_form_min` for aligned reads or `dio_rd_ill_min` for misaligned reads. By default, *value*=0, which means that no default direct reads occur based on I/O sizes.

For more information, see one or more of the following commands or mount parameters:

- [“The `:dio_rd_form_min eq value` and `:dio_wr_form_min eq value` Commands” on page 315](#)
- [“The `:dio_rd_ill_min eq value` and `:dio_wr_ill_min eq value` Commands” on page 315](#)

The `:dio_szero eq` and `:nodio_szero eq` Commands

These commands set or clear the direct I/O sparse zeroing mount option.

The `dio_szero` option causes uninitialized areas of sparse files written with direct I/O to be zeroed when the area is accessed. This 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`.

For *eq*, specify the Equipment Ordinal for the file system.

The `:forcedirectio eq` and `:noforcedirectio eq` 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 enables the default, which is buffered I/O.

For *eq*, specify the Equipment Ordinal for the file system.

When direct I/O is specified, the system transfers data directly between the user's buffer and disk. Use direct I/O only if the file system is used for large, block-aligned, sequential I/O.

For more information about I/O, see [“Advanced Topics” on page 185](#).

File System Commands: Sun StorageTek QFS Shared File Systems

The following file system commands are supported on Sun StorageTek QFS shared file systems only.

The `:meta_timeo eq interval` Command

The `metatimeo` command sets the Sun StorageTek QFS shared file system metadata cache timeout value. For more information about using this feature, see [“Retaining Cached Attributes: the `meta_timeo=n` Option” on page 257](#).

For *eq*, specify the Equipment Ordinal of the file system.

For *interval*, specify an interval in seconds. The default *interval* is 3. After this interval expires, the client host systems obtain a new copy of the metadata information from the metadata server host.

The `:mh_write eq` and `:nomh_write eq` Commands

These commands enable or disable multihost reads and writes. For information about this feature, see [“Enabling Multiple Host Reads and Writes: the `mh_write Option`” on page 255](#).

For *eq*, specify the Equipment Ordinal of the file system.

The `:minallocsz eq value` and `:maxallocsz eq value` Commands

These commands set the minimum and maximum block allocation size.

For *eq*, specify the Equipment Ordinal of the file system.

For *value*, and for more information about this feature, see [“Tuning Allocation Sizes: the `minallocsz=n` and `maxallocsz=n` Options” on page 253](#).

The `:rdlease eq interval`, `:wrlease eq interval`, and `:aplease eq interval` Commands

These commands tune the amount of time granted for read, write, and append leases. For information about this feature, see [“Using Leases in a Sun StorageTek QFS Shared File System: the `rdlease=n`, `wrlease=n`, and `aplease=n` Options” on page 254](#).

For *eq*, specify the Equipment Ordinal of the file system.

For *interval*, specify an integer number of seconds. All three leases enable you to specify an *interval* such that $15 \leq interval \leq 600$. The default *interval* is 30.

File System Commands: Miscellaneous

The following commands enable you to control leases, allocation sizes, and various other file system characteristics.

The `:abr eq` and `:noabr eq` Commands

These commands set or clear the application binary recovery (ABR) mount option.

For use in an Oracle RAC environment with Sun StorageTek QFS asynchronous I/O (AIO) only. These mount options disable or enable ABR of software mirrors. They apply only to Sun StorageTek QFS file systems built on Solaris Volume Manager mirrored volumes that support ABR.

For *eq*, specify the Equipment Ordinal for the file system.

The `:dmr eq` and `:nodmr eq` Commands

These commands set or clear the direct mirror reads (DMR) mount option.

For use in an Oracle Real Application Cluster (RAC) environment with Sun StorageTek QFS AIO only. These mount options disable or enable DMR of software mirrors. They apply only to Sun StorageTek QFS file systems built on Solaris Volume Manager mirrored volumes that support DMR.

For *eq*, specify the Equipment Ordinal for the file system.

The `:invalid eq interval` Command

The `invalid` command specifies that the file system hold cached attributes for at least *interval* seconds after a file is modified. You can specify this command only if the file system was mounted originally with the `reader` mount option. For information about mount options, see the `mount_samfs(1M)` man page.

For *eq*, specify the Equipment Ordinal for the file system.

For *interval*, specify the number of seconds to hold the attributes after file modification. For example, assume that *interval*=30. In such a file system, if you issue an `ls(1)` command, you might not see a newly created file appear in its output for 30 seconds after it has been created on its writer host.

The `:mm_stripe eq value` Command

The `mm_stripe` command sets the metadata stripe width for the file system to *value* 16-kilobyte disk allocation units (DAUs).

For *eq*, specify the Equipment Ordinal of the file system.

For *value*, specify either 0 or 1. If *value*=1, which is the default, the file system writes one DAU of metadata to one LUN before switching to another LUN. If *value*=0, the metadata is round-robin across all available metadata LUNs.

The `:qwrite eq` and `:noqwrite eq` Commands

The `qwrite` and `noqwrite` commands control the ability to perform simultaneous reads and writes to the same file from different threads. Specify `qwrite` only if file system users handle multiple simultaneous transactions to the same file. For example, this is useful in database applications. The `qwrite` feature improves I/O performance by queuing multiple requests at the drive level. The `qwrite` specification is disabled for NFS reads or writes of the file system.

The default setting is `noqwrite`, so the file system disables simultaneous reads and writes to the same file. This is the mode defined by the UNIX `vnode` interface standard that gives exclusive access to only one writer and forces other writers and readers to wait.

For *eq*, specify the Equipment Ordinal of the file system.

The `:refresh_at_eof eq` and `:norefresh_at_eof eq` Commands

The `refresh_at_eof` and `norefresh_at_eof` commands can be used for fast updates to Sun StorageTek 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, for example, if the writer host system is appending to a file and the reader is issuing `tail(1)` commands with the `-f` option. The default is `norefresh_at_eof`.

For *eq*, specify the Equipment Ordinal of the file system.

The `:suid eq` and `:nosuid eq` 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 see the `suid(2)` man page.

For *eq*, specify the Equipment Ordinal of the file system.

The `:stripe eq value` Command

The `stripe` command sets the stripe width for the file system to *value* disk allocation units (DAUs). The stripe width specifies that *value* multiplied by the DAU bytes are written to one LUN before switching to the next LUN. You can use the `sammkfs(1M) -a` command to set the DAU size on the file system when it is initialized.

For *eq*, specify the Equipment Ordinal of the file system.

For *value*, specify an integer such that $0 < value < 255$. If *value*=0, files are round-robin on each slice. The default *value* on file systems with an `ms` Equipment Type and on file systems with an `ma` Equipment Type with no striped group (`gXXX`) components is as follows:

- 128 kilobytes/DAU for DAUs < 128 kilobytes
- 1 for DAUs > 128 kilobytes

By default, *value*=0 on a Sun StorageTek 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 [“Design Basics” on page 7](#) and [“Configuring the File System” on page 29](#).

The `:sync_meta eq value` Command

The `sync_meta` command determines whether metadata is written to disk every time it changes. If you are using this command on Sun StorageTek QFS shared file system, also see [“Specifying the Frequency With Which Metadata Is Written: the `sync_meta=n` Option” on page 258](#).

For *eq*, specify the Equipment Ordinal of the file system.

For *value*, specify either 0 or 1, as follows:

- If *value* is 0, metadata is held in a buffer after it changes. For an unshared Sun StorageTek QFS file system in which higher performance is desired, you can set *value* to 0. In this case, the system performs a delayed write in which metadata is held in a buffer before it is written to disk. This is the default for unshared file systems and for file systems that are not mounted as multireader file systems.
- If *value* is 1, metadata is written to disk every time it changes. This slows performance, but it increases data consistency. This is the default for Sun StorageTek QFS file systems mounted as multireader file systems or as shared file systems. For a Sun StorageTek QFS shared file system, *value* must be set to 1 if failover capability is required.

The `:trace eq` and `:notrace eq` Commands

The `trace` command enables tracing for a file system. The `notrace` command disables tracing. These are global directives that affect all operations. For more information about file system tracing, see the `defaults.conf(4)` man page.

For `eq`, specify the Equipment Ordinal of a file system.

Miscellaneous Commands

The following commands enable you to control tracing, open access to a disk device, and perform several other miscellaneous tasks.

The `:clear vsn [index]` Command

The `clear` command clears the specified VSN from the removable media mount requests display. For more information, see [“\(p\) - Removable Media Load Requests Display” on page 287](#).

For `vsn`, specify the volume to mount. Any process waiting for the VSN mount is aborted.

For `index`, specify the decimal ordinal of the VSN in the removable media display.

The `:devlog eq [option]` Command

The `devlog` command sets one or more events to be logged.

For `eq`, specify the Equipment Ordinal of a device.

For `option`, specify one or more event types. Possible event types are as follows: `all`, `date`, `default`, `detail`, `err`, `event`, `label`, `mig`, `module`, `msg`, `none`, `retry`, `stage`, `syserr`, and `time`. For information about these options, see the `defaults.conf(4)` man page. If no `option` is specified, the system does not change the current events being logged for the `eq` specified.

The `:diskvols volume [+flag | -flag]` Command

The `diskvols` command sets or clears flags in the disk volume dictionary.

For `volume`, specify the volume in the disk volume dictionary.

For *flag*, specify one of the five flags in the `D samu(1M)` display. For information about the disk volume dictionary and the flags see “(D) - Disk Volume Dictionary” on page 270 or the `samu(1M)` man page.

The `:dtrace` Commands

The `dtrace` commands are as follows:

- `:dtrace daemon_name on`
- `:dtrace daemon_name off`
- `:dtrace daemon_name.variable value`

The `dtrace` commands specify various tracing options. TABLE C-40 shows the tracing control command arguments.

TABLE C-40 Tracing Command Arguments

Argument	Description
<i>daemon_name</i>	Specify the keyword <code>all</code> or a process name. If the keyword <code>all</code> is specified, the tracing command affects all daemons. If one of the following process names is specified, the tracing command affects that process only: <code>sam-archiverd</code> , <code>sam-catserverd</code> , <code>sam-fsd</code> , <code>sam-rftd</code> , <code>sam-recycler</code> , <code>sam-sharefsd</code> , and <code>sam-stagerd</code> . One of the keywords <code>on</code> or <code>off</code> can be specified after a process name. If <code>on</code> or <code>off</code> is specified, tracing is turned off or on for all processes specified.
<i>variable value</i>	Many different <i>variable</i> and <i>value</i> arguments can be specified. The <code>defaults.conf(4)</code> man page contains comprehensive information about these arguments. Specify one of the following <i>variable</i> and <i>value</i> combinations: <ul style="list-style-type: none"> • <i>file value</i>. For <i>value</i>, specify the name of a file to which trace files can be written. This can be a full path name. • <i>options value</i>. For <i>value</i>, specify a space-separated list of trace options. • <i>age value</i>. For <i>age</i>, specify the trace file rotation age. Note: Do not set this value to two minutes or less. If you do, the rotation will never take place. • <i>size value</i>. For <i>value</i>, specify the size of the trace file at which rotation will begin.

The `:fs fsname` Command

The `fs` command sets the file system to be displayed through the `N` display.

For *fsname*, specify the name of the file system to be examined.

The `:mount mntpt` Command

The `mount` command selects a Sun StorageTek QFS file system. For *mntpt*, specify the mount point of a file system.

The `:open eq` Command

The `open` command enables access to the specified disk device. You must issue this command before you can use the `read` command, disk sector display (S), or file label display (F).

For *eq*, specify the Equipment Ordinal of a device.

The `:read addr` Command

The `read` command reads the specified sector from the currently opened disk device. You must open the device before it can be read.

For *addr*, specify the hexadecimal sector address.

The `:refresh i` Command

The `refresh` command determines the amount of time between `samu(1M)` screen refreshes.

For *i*, specify a time in seconds.

The `:snap [filename]` Command

The `snap` command sends a snapshot of a display window to *filename*, which is the name of a file to receive the display information.

To aid in problem reporting, you can take a snapshot of all the `samu(1M)` utility's displays. Each new snapshot is appended to the snapshots file. The default file is `snapshots` in the current working directory. The file can be printed, examined using `vi(1)`, or faxed to Sun Microsystems customer support staff.

The `:! shell_command` Command

The `!` command enables you to run a shell command without leaving the `samu(1M)` operator utility.

Glossary

A

- addressable storage** The storage space encompassing online, nearline, offsite, and offline storage that is user-referenced through a Sun StorageTek QFS or Sun StorageTek SAM file system.
- archive media** The media to which an archive file is written. Archive media can be removable tape or magneto-optical cartridges in a library. In addition, archive media can be a mount point on another system.
- archiver** The archive program that automatically controls the copying of files to removable cartridges.
- archive storage** Copies of file data that have been created on archive media.
- audit (full)** The process of loading cartridges to verify their VSNs. For magneto-optical cartridges, the capacity and space information is determined and entered into the automated library's catalog. See also, *VSN*.
- automated library** A robotically controlled device designed to automatically load and unload removable media cartridges without operator intervention. An automated library contains one or more drives and a transport mechanism that moves cartridges to and from the storage slots and the drives.

B

- backup storage** A snapshot of a collection of files for the purpose of preventing inadvertent loss. A backup includes both the file's attributes and associated data.

block allocation map A bitmap representing each available block of storage on a disk and indicating whether the block is in use or free.

block size See *DAU*.

C

cartridge A physical entity that contains media for recording data, such as a tape or optical disk. Sometimes referred to as *a piece of media, a volume, or the medium*.

catalog A record of the VSNs in an automated library. There is one catalog for each automated library and, at a site, there is one historian for all automated libraries. See also, *VSN*.

client-server The model of interaction in a distributed system in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called the client. The program satisfying the response is called the server.

connection The path between two protocol modules that provides reliable stream delivery service. A TCP connection extends from a TCP module on one machine to a TCP module on the other.

D

data device In a file system, a device or group of devices upon which file data is stored.

DAU Disk allocation unit. The basic unit of online storage. Also called *block size*.

device logging A configurable feature that provides device-specific error information used to analyze device problems.

device scanner Software that periodically monitors the presence of all manually mounted removable devices and that detects the presence of mounted cartridges that can be requested by a user or other process.

direct access A file attribute (stage never) designating that a nearline file can be accessed directly from the archive media and need not be retrieved to disk cache.

direct attached library An automated library connected directly to a server using a SCSI interface. A SCSI-attached library is controlled directly by the Sun StorageTek SAM software.

- direct I/O** An attribute used for large block-aligned sequential I/O. The `setfa(1)` command's `-D` option is the direct I/O option. It sets the direct I/O attribute for a file or directory. If applied to a directory, the direct I/O attribute is inherited.
- directory** A file data structure that points to other files and directories within the file system.
- disk allocation unit** See *DAU*.
- disk buffer** In a Sun SAM-Remote configuration, the buffer on the server system that is used for archiving data from the client to the server.
- disk cache** The disk-resident portion of the file system software, used to create and manage data files between online disk cache and archive media. Individual disk partitions or an entire disk can be used as disk cache.
- disk space threshold** The maximum or minimum level of disk cache utilization, as defined by an administrator. The releaser controls disk cache utilization based on these predefined disk space thresholds.
- disk striping** The process of recording a file across several disks, thereby improving access performance and increasing overall storage capacity. See also *striping*.
- drive** A mechanism for transferring data to and from a removable media volume.

E

- Ethernet** A local-area, packet-switched network technology. Originally designed for coaxial cable, it is now found running over shielded, twisted-pair cable. Ethernet is a 10- or 100-Mbytes/second LAN.
- extent array** The array within a file's inode that defines the disk location of each data block assigned to the file.

F

- family device set** See *family set*.
- family set** A storage device that is represented by a group of independent physical devices, such as a collection of disks or the drives within an automated library. See also *storage family set*.

- FDDI** Fiber-distributed data interface (FDDI) is a standard for data transmission in a local area network that can extend in range up to 200 km (124 miles). The FDDI protocol is based on the token ring protocol.
- Fibre Channel** The ANSI standard that specifies high-speed serial communication between devices. Fibre Channel is used as one of the bus architectures in SCSI-3.
- file system** A hierarchical collection of files and directories.
- file-system-specific directives** Archiver and releaser directives that follow global directives in the `archiver.cmd` file, are specific to a particular file system, and begin with `fs =`. File-system-specific directives apply until the next `fs =` directive line or the end of file is encountered. If multiple directives affect a file system, the file-system-specific directives override the global directives.
- FTP** File transfer protocol. An Internet protocol for transferring files between two hosts over a TCP/IP network.
-

G

- global directives** Archiver and releaser directives that apply to all file systems and that appear before the first `fs =` line.
- grace period** For disk quotas, the amount of time for which a user is allowed to create files and allocate storage after reaching the soft limit.
-

H

- hard limit** For disk quotas, the maximum limit on file system resources, blocks, and inodes that users cannot exceed.
- hosts file** The hosts file contains a list of all of the hosts in a shared file system. If you are initializing a file system as a Sun StorageTek QFS shared file system, the hosts file, `/etc/opt/SUNWsamfs/hosts.fs-name`, must be created before the file system is created. The `sammkfs(1M)` command uses the hosts file when it creates the file system. You can use the `samsharefs(1M)` command to replace or update the contents of the hosts file at a later date.

I

- indirect block** A disk block that contains a list of storage blocks. File systems have up to three levels of indirect blocks. A first-level indirect block contains a list of blocks used for data storage. A second-level indirect block contains a list of first-level indirect blocks. A third-level indirect block contains a list of second-level indirect blocks.
- inode** Index node. A data structure used by the file system to describe a file. An inode describes all the attributes associated with a file other than the name. The attributes include ownership, access, permission, size, and the file location on the disk system.
- inode file** A special file (`.inodes`) on the file system that contains the inode structures for all files resident in the file system. Inodes are 512 bytes long. The inode file is a metadata file, which is separated from file data in the file system.

K

- kernel** The central controlling program that provides basic system facilities. The UNIX kernel creates and manages processes, provides functions to access the file system, provides general security, and supplies communication facilities.

L

- LAN** Local area network.
- lease** A function that grants a client host permission to perform an operation on a file for a specified period of time. The metadata server issues leases to each client host. The leases are renewed as necessary to permit continued file operations.
- library** See *automated library*.
- library catalog** See *catalog*.
- local file system** A file system that is installed on one node of a Sun Cluster system and is not made highly available to another node. Also, a file system that is installed on a server.
- LUN** Logical unit number.

M

- mc f** Master configuration file. The file that is read at initialization time that defines the relationships between the devices (the topology) in a file system environment.
- media** Tape or optical disk cartridges.
- media recycling** The process of recycling or reusing archive media with few active files.
- metadata** Data about data. Metadata is the index information used to locate the exact data position of a file on a disk. It consists of information about files, directories, access control lists, symbolic links, removable media, segmented files, and the indexes of segmented files.
- metadata device** A device (for example, a solid-state disk or mirrored device) upon which file system metadata is stored. Having file data and metadata on separate devices can increase performance. In the `mc f` file, a metadata device is declared as an `mm` device within an `ma` file system.
- mirror writing** The process of maintaining two copies of a file on disjointed sets of disks to prevent loss from a single disk failure.
- mount point** The directory on which a file system is mounted.
- multireader file system** A single-writer, multireader capability that enables you to specify a file system that can be mounted on multiple hosts. Multiple hosts can read the file system, but only one host can write to the file system. Multiple readers are specified with the `-o reader` option with the `mount(1M)` command. The single-writer host is specified with the `-o writer` option with the `mount(1M)` command. For more information on the `mount(1M)` command, see the `mount_samfs(1M)` man page.

N

- name space** The metadata portion of a collection of files that identifies the file, its attributes, and its storage locations.
- nearline storage** Removable media storage that requires robotic mounting before it can be accessed. Nearline storage is usually less expensive than online storage, but it takes somewhat longer to access.

**network attached
automated library**

A library, such as those from StorageTek, ADIC/Grau, IBM, or Sony, that is controlled using a software package supplied by the vendor. The Sun StorageTek SAM file system interfaces with the vendor software using a Sun StorageTek SAM media changer daemon designed specifically for the automated library.

NFS Network file system. A file system distributed by Sun that provides transparent access to remote file systems on heterogeneous networks.

NIS The Sun OS 4.0 (minimum) Network Information Service. A distributed network database containing key information about systems and users on the network. The NIS database is stored on the master server and all slave servers.

O

offline storage Storage that requires operator intervention for loading.

offsite storage Storage that is remote from the server and is used for disaster recovery.

online storage Storage that is immediately available, such as disk cache storage.

P

partition A portion of a device or a side of a magneto-optical cartridge.

preallocation The process of reserving a contiguous amount of space on the disk cache for writing a file. Preallocation can be specified only for a file that is size zero. For more information, see the `setfa(1)` man page.

pseudo device A software subsystem or driver with no associated hardware.

Q

quota The amount of system resources that a user is allowed to consume.

R

- RAID** Redundant array of independent disks. A disk technology that uses several independent disks to reliably store files. It can protect against data loss from a single disk failure, can provide a fault-tolerant disk environment, and can provide higher throughput than individual disks.
- recycler** A Sun StorageTek SAM utility that reclaims space on cartridges that is occupied by expired archive copies.
- release priority** The priority according to which a file in a file system is released after being archived. Release priority is calculated by multiplication of various weights of file properties and then summation of the results.
- releaser** A Sun StorageTek SAM component that identifies archived files and releases their disk cache copies, thus making more disk cache space available. The releaser automatically regulates the amount of online disk storage according to high and low thresholds.
- remote procedure call** See *RPC*.
- removable media file** A special type of user file that can be accessed directly from where it resides on a removable media cartridge, such as magnetic tape or optical disk cartridge. Also used for writing archive and stage file data.
- robot** The portion of an automated library that moves cartridges between storage slots and drives. Also called a *transport*.
- round-robin** A data access method in which entire files are written to logical disks in a sequential fashion. When a single file is written to disk, the entire file is written to the first logical disk. The second file is written to the next logical disk, and so on. The size of each file determines the size of the I/O.
- See also *disk striping* and *striping*.
- RPC** Remote procedure call. The underlying data exchange mechanism used by NFS to implement custom network data servers.

S

- samfsdump** A program that creates a control structure dump and copies all the control structure information for a given group of files. It is analogous to the UNIX *tar(1)* utility, but it does not generally copy file data. See also *samfsrestore*.

SAM-QFS	A configuration that combines the Sun StorageTek SAM software with the Sun StorageTek QFS file system. SAM-QFS offers a high-speed, standard UNIX file system interface to users and administrators in conjunction with the storage and archive management utilities. It uses many of the commands available in the Sun StorageTek SAM command set as well as standard UNIX file system commands.
<code>samfsrestore</code>	A program that restores inode and directory information from a control structure dump. See also <i>samfsdump</i> .
SCSI	Small Computer System Interface. An electrical communication specification commonly used for peripheral devices such as disk and tape drives and automated libraries.
shared hosts file	When you create a shared file system, the system copies information from the hosts file to the shared hosts file on the metadata server. You update this information when you issue the <code>samsharefs(1M) -u</code> command
Small Computer System Interface	See SCSI.
soft limit	For disk quotas, a threshold limit on file system resources (blocks and inodes) that you can temporarily exceed. Exceeding the soft limit starts a timer. When you exceed the soft limit for the specified time, no further system resources can be allocated until you reduce file system use to a level below the soft limit.
staging	The process of copying a nearline or offline file from archive storage back to online storage.
storage family set	A set of disks that are collectively represented by a single disk family device.
storage slots	Locations inside an automated library in which cartridges are stored when not being used in a drive.
striped group	A collection of devices within a file system that is defined in the <code>mcf</code> file as one or more <code>gXXX</code> devices. Striped groups are treated as one logical device and are always striped with a size equal to the disk allocation unit (DAU).
stripe size	The number of disk allocation units (DAUs) to be allocated before writing proceeds to the next device of a stripe. If the <code>stripe=0</code> mount option is used, the file system uses round-robin access, not striped access.
striping	A data access method in which files are simultaneously written to logical disks in an interlaced fashion. SAM-QFS file systems provide two types of striping: "hard striping," using stripe groups, and "soft striping," using the <code>stripe=x</code> mount parameter. Hard striping is enabled when a file system is set up, and requires the definition of stripe groups within the <code>mcf</code> file. Soft striping is enabled through the <code>stripe=x</code> mount parameter, and can be changed for the file system or for individual files. It is disabled by setting <code>stripe=0</code> . Hard and soft striping can both be used if a file system is composed of multiple stripe groups with the same number of elements. See also <i>round-robin</i> .

- Sun SAM-Remote client** A Sun StorageTek SAM system with a client daemon that contains a number of pseudodevices, and can also have its own library devices. The client depends on a Sun SAM-Remote server for archive media for one or more archive copies.
- Sun SAM-Remote server** Both a full-capacity Sun StorageTek SAM storage management server and a Sun SAM-Remote server daemon that defines libraries to be shared among Sun SAM-Remote clients.
- superblock** A data structure in the file system that defines the basic parameters of the file system. The superblock is written to all partitions in the storage family set and identifies the partition's membership in the set.
-

T

- tar** Tape archive. A standard file and data recording format used for archive images.
- TCP/IP** Transmission Control Protocol/Internet Protocol. The internet protocols responsible for host-to-host addressing and routing, packet delivery (IP), and reliable delivery of data between application points (TCP).
- timer** Quota software that keeps track of the period starting when a user reaches a soft limit and ending when the hard limit is imposed on the user.
-

V

- vfstab file** The `vfstab` file contains mount options for the file system. Mount options specified on the command line override those specified in the `/etc/vfstab` file, but mount options specified in the `/etc/vfstab` file override those specified in the `samfs.cmd` file.
- volume** A named area on a cartridge for sharing data. A cartridge has one or more volumes. Double-sided cartridges have two volumes, one on each side.
- volume overflow** A capability that enables the system to span a single file over multiple volumes. Volume overflow is useful for sites using very large files that exceed the capacity of their individual cartridges.

VSN Volume serial name. In the context of archiving to removable media cartridges, the VSN is a logical identifier for magnetic tape and optical disk that is written in the volume label. In the context of archiving to disk cache, this is the unique name for the disk archive set.



W

WORM Write once read many. A storage classification for media that can be written only once but read many times.

Index

Symbols

! samu(1M) command, 323
.inodes file, 3

Numerics

64-bit addressing, 3

A

abr mount option, 125
abr samu(1M) command, 318
additional parameters field, 38
admin set, quotas, 97, 104
alloc command, 28
allocation
 nalloc and alloc commands, 28
API routines, 6, 50
aplease
 mount option, 254
 samu(1M) command, 317
archive
 copy, 53
archive(1) command, 49
archiver
 samu(1M) display, 263
automated library
 samu(1M) display, 287, 300

B

backing up files, 66
bg mount option, 252
block quotas, SAN-QFS file system, 217

Buffered I/O
 see I/O, paged

C

cached
 I/O, see I/O, paged
chmod 4000 WORM trigger, 196
chmod -w WORM trigger, 198
clear samu(1M) command, 321
clients outside the cluster, 165
client-server communication, 94
configuration
 changes, propagating, 54
 creating the mcf file, 34
 disk use, 8
 file allocation, 14
 samu(1M) display, 265

D

daemons
 overview, 186
 sam-catservrd, 186
 sam-fsd, 80, 186
 sam-genericd, 186
 sam-rpcd, 186
 sam-sharedfsd, 186
 sam-sharefsd, 80, 186
 samu(1M) display, 268
 shared file system, 186
 tracing, 187
 tracing, samu(1M) command, 322
 xntpd(1M), 244

- data alignment, 11
- DAU, 6
 - dual allocation scheme, 9
 - md device, 11
 - mr or gXXXX, 10
 - overview, 8
 - quotas and DAUs, 99
 - settings, 9
 - single allocation scheme, 10
- dd(1M) command, 101, 103
- def_retention mount option, 258
- defaults.conf file
 - enabling tracing, 244
 - propagating changes, 55
- device commands in samu(1M), 311
- device state codes, viewing with samu(1M), 308
- device state field, 38
- device status, samu(1M) display, 292
- device table, samu(1M) display, 299
- devlog samu(1M) command, 321
- DID support, 124
- dio_rd_consec
 - mount option, 220
 - samu(1M) command, 315
- dio_rd_form_min samu(1M) command, 315
- dio_rd_ill_min samu(1M) command, 315
- dio_szero samu(1M) command, 316
- dio_wr_consec
 - mount option, 220
 - samu(1M) command, 315
- dio_wr_form samu(1M) command, 315
- dio_wr_ill_min samu(1M) command, 315
- directio(3C) function call, 3, 219
- directory attributes, 190
- directory name lookup cache (DNLC), 7, 95
- disk allocation unit, see DAU
- disk cache
 - adding, 64
 - exceeding, 207
- disk volume dictionary
 - samu(1M) display, 270
- disks
 - adding or upgrading, 66
- diskvols samu(1M) command, 321
- diskvols.conf file

- samu(1M) display, 270
- dmr mount option, 125
- dmr samu(1M) command, 318
- documentation, xxx
- down samu(1M) command, 311
- dtrace samu(1M) command, 322
- dual allocation scheme, 9

E

- EDOM error, 229
- EDQUOT error, 99
- EFI labels
 - on a Linux client, 248
- emul_lite mount option (WORM), 193
- emulation mode, WORM, 193
- ENOCSE error, 229
- Enterprise Volume Management System (EVMS), 248
- equipment
 - identifier field, 35
 - ordinal field, 36
 - type field, 36

F

- family set field, 37
- file allocation
 - methods, 191
 - mismatched striped groups, 23
 - preallocating space, 190
 - round-robin, 14, 15
 - striped, 14, 17
 - striped groups, 20
- file attributes, 7, 49, 190
- file retention, 54
- file system
 - adding disk cache, 64
 - backing up, 66
 - basic operations, 49
 - capacity, 3
 - commands, using samu(1M), 312, 317
 - converting to shared, 71
 - design, 7
 - direct I/O commands, using samu(1M), 314
 - exceeding cache, 207
 - features, 2
 - file allocation, 14

- file information, 51
- ma, 36
- metadata, 4
- mounting, 60
- ms, 36
- ms and ma, 9
- number of files supported, 3
- parameters, `samu(1M)` display, 284
- quotas, see quotas
- recovery, 3
- recreating, 66
- renaming, 65
- repair, 231
- `samu(1M)` display, 271
- shared, see shared file system
- status codes, viewing with `samu(1M)`, 307
- unmounting, 63
- usage, `samu(1M)` display, 278
- volume management, 2

File System Manager

- adding a managed server, 33
- assigning privilege levels, 31
- creating accounts, 30
- overview, 29
- session timeout setting, 34
- using, 30

files

- accommodating large files, 207
- large file transfer, 220

`flush_behind`

- mount option, 225
- `samu(1M)` command, 312

`force_nfs_async` `samu(1M)` command, 312

forced unmount, 64

`forcedirectio`

- mount option, 219
- `samu(1M)` command, 316

`fs` `samu(1M)` command, 322

`fsck(1M)` command, also see `samfsck(1M)`

- command, 3

`fsmadm(1M)` command, 33

G

- grace period for quotas, 111, 113
- `gXXX` devices, 37

H

HA-NFS

- configuration, 151, 164
- disabling, 178, 179, 181

hard limit, quotas, 99

HA-SAM, 183

hash table, 227

HASStoragePlus resource type, 144, 152

help

- `samu(1M)` display, 273

high-availability SAM configuration, 183

`hostname(1M)` command, 243

hosts file, 73, 78

- deleting a host, 83
- example, 88
- with Sun Cluster, 140

`hosts.fsnname.local` file, 74, 87, 88

I

I/O

- direct, 2, 219
- direct I/O file attribute, 50
- direct I/O `samu(1M)` commands, 314
- paged, 2, 221, 222
- switching, 220
- tuning, 219

idle `samu(1M)` command, 311

infinite quotas, 105

inode

- attributes, 49
- file content, 7
- `samu(1M)` display, 274

inodes file, 3

inodes hash table, tuning, 226

invalid `samu(1M)` command, 318

IPMP

- adapters, 151
- validation testing, 149

K

- kernel statistics, `samu(1M)` display, 277

L

- leap year, WORM retention periods, 202
- leases, 254

library catalog, samu(1M) display, 300

licensing

 general information, xxxii

Linux client, 5

 frequently asked questions, 248

 log files, 246

 mount options, 249

 OS versions supported, xxvii

 troubleshooting, 246

local hosts file, 74, 80

 creating, 87

log files

 file system troubleshooting, 230

 Linux client, 246

 sam-sharesfd trace log, 244, 245

logical volume management (LVM), 248

LogicalHost resource type, 179

ls(1) command, also see sls(1) command, 51

LUN

 control allocation by LUN, 28

M

ma file system, 9, 36

mass-storage, samu(1M) display, 280

maxallocsz

 mount option, 253

 samu(1M) command, 317

maxphys parameter, 220

mcf file

 /dev/dsk entry, 35

 /dev/rmt entry, 36

 /dev/samst entry, 35

 additional parameters field, 38

 configuring, 34

 device state field, 38

 equipment identifier field, 35

 equipment ordinal field, 36

 equipment type field, 36

 error checking, 67

 examples, 38, 42

 family set field, 37

 fields, 34

 increasing file system size, 65

 propagating changes, 55

 shared file system, 84

 shared file system on Sun Cluster, 137

 shared keyword, 72

 Sun Cluster sample, 156

 unshared file system in Sun Cluster, 147

 updating, 84

 verifying, 232

 with Sun Cluster, 162

md device, 36

mdadm, not to be used on Linux client, 249

memory, samu(1M) display, 267

messages file, 65, 230

meta_timeo

 mount option, 257

 samu(1M) command, 316

metadata, 7

 allocation, 14

 content, 7

 device in mcf, 36

 overview, 4

 separation, 7

 server, 4

metadata server, see Shared file system

metadb(1M) command, 154

metaset command, 157

mh_write

 mount option, 255

 samu(1M) command, 317

min_pool mount option, 256

minallocsz

 mount option, 253

 samu(1M) command, 317

mm device, 36

mm_stripe samu(1M) command, 318

mount options

 abr, 125

 bg, 252

 dmr, 125

 flush_behind, 225

 for quotas, 98

 Linux client, 249

 meta_timeo, 257

 mh_write, 255

 min_pool, 256

 minallocsz and maxallocsz, 253

 rdlease, wrlease, and aplease, 254

 retry, 252

 shared, 253

 shared file system, 251

 stage_flush_behind, 225

- stripe, 257
- sync_meta, 258
- worm_capable and def_retention, 258
- WORM-FS, 194
- wr_throttle, 224
- mount parameters, 60
- mount point, verifying, 233
- mount samu(1M) command, 323
- mount(1M) command, 40, 60, 61
 - failure, 232
 - shared file system, 70
 - troubleshooting, 95, 231, 238
- mr device, 36
- ms file system, 9, 36
- multireader file system, 208

N

- nalloc command, 28
- netstat(1M) command, 238, 239
- Network Appliance interface emulation, 193
- network name service, 149
- NFS
 - resource group, 159
 - share point, 152
- nhino parameter, 226, 227
- ninodes parameter, 226, 227
- noabr samu(1M) command, 318
- nodev keyword, 35
- nodio_szero samu(1M) command, 316
- nodmr samu(1M) command, 318
- noforce_nfs_async samu(1M) command, 312
- noforcedirectio samu(1M) command, 316
- nomh_write samu(1M) command, 317
- noquota mount option, 117
- noqwrite samu(1M) command, 319
- norefresh_at_eof samu(1M) command, 319
- nosetuid samu(1M) command, 319
- nosw_raid samu(1M) command, 313
- notrace samu(1M) command, 321
- nsswitch.conf file, 149
- nstreams mount option, 256

O

- OCR file, 143

- off samu(1M) command, 311
- on samu(1M) command, 311
- open samu(1M) command, 323
- operating system, xxvii
- operator utility, see samu(1M)
- optical disk label, samu(1M) display, 273
- optical disk status
 - samu(1M) display, 286
- Oracle Cluster Registry (OCR) file, 143
- Oracle RAC, 138
 - with Sun Cluster, 142

P

- paged I/O, see I/O
- partitions (adding, changing, deleting), 66
- pending stage queue, samu(1M) display, 304
- per-LUN allocation control, 28
- preallocation of file space, 6, 190
- processes, see daemons
- propagating configuration changes, 54

Q

- qfsrestore(1M) command, 68
- quota mount option, 98
- quota.group file, 107
- quotas
 - admin set, 97, 104
 - block quotas in SAN-QFS, 217
 - changing, 111
 - changing limits, 107
 - checking, 109
 - commands, 100
 - configuring, 100, 102
 - correcting, 118
 - DAU with quotas, 99
 - default, 106
 - disk blocks and file limits, 99
 - enabling, 100
 - enabling limits, 107
 - grace period, 111, 113
 - hard limit, 99
 - infinite, 105
 - inhibiting resource allocation, 115
 - overview, 97
 - quota files, 98
 - removing, 111, 117

- soft limit, 99
- viewing group quotas, 116
- zero, 105

Qwrite, 223

qwrite

- mount option, 224
- samu(1M) command, 319

R

rdlease

- mount option, 254
- samu(1M) command, 317

read samu(1M) command, 323

readahead

- mount option, 222
- samu(1M) command, 313

reader mount option, 208

recovery of a file system, 3

Red Hat Linux, supported distributions, 5

refresh samu(1M) command, 323

refresh_at_eof samu(1M) command, 319

release(1) command, 49

removable media

- samu(1M) display, 287
- status codes, samu(1M), 306
- status, samu(1M) display, 290

removing quotas, 117

renaming a file system, 65

repairing a file system, 231

retention, 54

- default period, WORM-FS, 202
- mount options, 258
- period, WORM-FS, 200
- setting the period using touch, 203

retry mount option, 252

round-robin allocation, 15, 36

- example, 42
- user specified, 191

S

sam_archive(3) API routine, 50

sam_release(3) API routine, 50

sam_segment(3) API routine, 50

sam_setfa(3) API routine, 50, 219

sam_ssum(3) API routine, 50

sam_stage(3) API routine, 50

Samba, 213

sam-catsvervd daemon, 186

samchaid(1M) command, 105

samcmd(1M) aridle command, 57

samcmd(1M) command, 60

samd(1M) command, 57, 65, 67

samd(1M) config command, 73

samfs.cmd file, 40, 60, 220

- example, 62
- with Sun Cluster, 138

samfsck(1M) command, 37, 72, 76, 230, 231

samfsconfig(1M) command, 236

- example output, 85

sam-fsd daemon, 80, 186

sam-fsd(1M) command, 65, 67, 188

samfsinfo(1M) command, 41

- example output, 235

samfsrestore(1M) command, 68

sam-genericd daemon, 186

samgrowfs(1M) command, 64, 66, 218

sammkfs(1M) command, 27, 37, 41, 67

- troubleshooting, 231

SAM-QFS, xxvii

samquota(1M) command, 100

- arguments, 112

SAM-Remote, samu(1M) display, 292

sam-rpcd daemon, 186

sam-sharedfsd daemon, 186

samsharefs(1M) command, 58, 73, 78

- example output, 236

sam-sharefsd daemon, 80, 186

- troubleshooting, 95

sam-sharefsd trace log, 244

samu(1M) utility

- device commands, 311
- direct I/O commands, 314
- display keys, 261
- file system commands, 312, 317
- help display, 273
- interface, 261
- invocation, 259
- mount command, 60
- nalloc and alloc commands, 28
- operator commands, 311

- operator displays, 262
- overview, 259
- shared file system commands, 316
- status codes, 306
- samunhold(1M) command, 217
- SANergy, 212
 - file holds, 217
 - installing, 214
- SANERGY_SMBPATH environment variable, 213
- SAN-QFS file system
 - clients, 213
 - comparison to Sun StorageTek QFS shared file system, 218
 - enabling, 212
 - expanding, 218
 - limitations, 212
 - metadata controller, 212
 - overview, 210
 - troubleshooting, 217
 - unmounting, 215
- sconf command, 162, 166
- scdidadm(1M) command, 127, 161
- scrgadm(1M) command, 132, 148, 152, 158
- SCSI disk, transfers, 221
- SCSI status, samu(1M) display, 296
- scswitch(1M) command, 132, 142, 159
- sd.conf file, 221
- sd_max_xfer_size definition, 221
- sector data, samu(1M) display, 294
- segment(1) command, 50, 208
- services, samu(1M) display, 290
- setfa file attributes, 50
- setfa(1) command, 3, 27, 50, 190, 219
- setuid samu(1M) command, 319
- sfind(1M) command
 - WORM options, 206
- share(1M) command, 213
- shared file system, 5
 - adding a client host, 77
 - aplease mount option, 254
 - cached attributes, 257
 - changing the metadata server, 90, 93
 - clients outside Sun Cluster, 165
 - client-server communication, 94
 - commands, using samu(1M), 316
 - comparison to SAN-QFS shared file system, 218
 - configuring, 69
 - converting to unshared, 75
 - converting unshared to shared, 71
 - daemons, 186
 - example with Sun Cluster, 135
 - failover, 90, 91, 93
 - file locking, 5
 - hosts file, 73, 78
 - I/O, 94
 - leases, 254
 - Linux client, 5
 - local hosts file, 74, 80
 - maxallocsz mount option, 253
 - mcf file, 84
 - meta_timeo mount option, 257
 - mh_write mount option, 255
 - min_pool mount option, 256
 - minallocsz mount option, 253
 - mount options, 251
 - mount points, 4
 - mounting, 70
 - nodev keyword, 35
 - removing a client host, 82
 - sam-fsd daemon, 80
 - sam-sharefsd daemon, 80
 - stripe mount option, 257
 - sync_meta mount option, 258
 - unmounting, 82
 - with Sun Cluster, 134, 139
 - wrlease mount option, 254
- shared hosts file
 - changing, 58, 59
- shared memory, samu(1M) display, 276, 279, 282
- shared mount option, 253
- single allocation scheme, 10
- sls(1) command, 49, 51
 - output explanation, 51
 - WORM options, 205
- snap samu(1M) command, 323
- soft limit, quotas, 99
- software
 - documentation, xxx
 - licensing, xxxii
 - operating system requirements, xxvii
- Solaris Volume Manager for Sun Cluster, 125
- Solstice DiskSuite/Solaris Volume Manager, 153
- squota(1) command, 100

- ssd.conf file, 221
- ssd_max_xfer_size definition, 221
- ssum(1) command, 49
- stage(1) command, 50
- stage_flush_behind
 - mount option, 225
- stager
 - pending stage queue samu(1M) display, 304
 - status, samu(1M) display, 283, 297
- status codes
 - viewing with samu(1M), 306, 307
- stripe
 - mount option, 223, 257
 - samu(1M) command, 320
- stripe widths
 - data disks, 11
 - metadata disks, 14
- striped allocation, 17, 36, 257
 - example, 44
 - stripe width, 11, 191, 223
 - user specified, 191
- striped group, 6, 20, 37
 - device, 192
 - example, 45
 - file attribute, 50
 - mismatched, 23
- striping
 - introduction, 17
- Sun Cluster
 - changing the configuration, 177
 - clients outside the cluster, 165
 - configuration examples, 132
 - configuring an unshared file system, 144
 - configuring clients outside the cluster, 165
 - DID support, 124
 - disabling HA-NFS, 178, 179, 181
 - documentation, 122
 - HA-NFS configuration, 151, 158, 164
 - HA-SAM configuration, 183
 - HASStoragePlus resource type, 144, 152
 - IPMP validation testing, 149, 158, 164
 - metadata server resource, 134
 - network name service, 149, 158, 164
 - NFS resource group, 152
 - Oracle RAC, 138, 142
 - overview, 122
 - private interconnect names, 139
 - Resource Group Manager, 144
 - restrictions, 123
 - shared file system, 134
 - shared file system example, 135
 - unshared file system example, 145
 - validating configuration, 140
 - with a shared or unshared file system, 124
 - with Solaris Volume Manager for Sun Cluster, 125
 - with Solstice DiskSuite/Solaris Volume Manager, 153
 - with VERITAS Clustered Volume manager, 160
- SUNW.qfs resource type, 132, 142
- superblocks, version 1 and 2, 11, 41
- SuSE Linux, supported distributions, 5
- sw_raid samu(1M) command, 313
- sync_meta
 - mount option, 258
 - samu(1M) command, 320
- syslog.conf file, 230
- system file, 220

T

- tape drive
 - samu(1M) display, 295
- technical support, xxxii
- threads in a shared file system, 256
- touch utility, setting the WORM-FS retention period, 203
- trace file, 187
 - directory, 187
 - events, 187
 - rotation, 188
- trace log files, 244
- trace samu(1M) command, 321
- trace_rotate(1M) command, 188
- tuning allocation sizes, 253

U

- unavail samu(1M) command, 311
- unfuse command, 215
- unload samu(1M) command, 312
- unloading media, with samu(1M), 312
- unmount(1M) command, 70
 - shared file system, 82
- unmounting

- file system, 63
- SAN-QFS file system, 215
- unshare(1M) command
 - SAN-QFS file system, 216
- upgrading
 - disks, 66

V

- VERITAS Clustered Volume manager, 160
- VFS, vnode interface
- vfstab file, 40, 60
 - fields, 61
 - retention period, WORM-FS, 202
 - shared file system, 81
 - shared keyword, 72
 - with Sun Cluster, 148, 157
- Vnode interface
 - see VFS
- volume management, 2
- vrdiskadd command, 161
- vrmpadm command, 161
- vxvg command, 162
- VxVM, 160
 - configuration, 161
 - devices, 182
 - packages, 160

W

- WORM retention, 54
- worm_capable mount option, 193, 258
- worm_emul mount option, 193
- worm_lite mount option, 193
- WORM-FS
 - "lite" options, 194
 - creating a WORM file, 195, 200
 - emulation mode, 193
 - mount options, 194, 258
 - options, 200
 - overview, 192
 - searching for, 206
 - setting the default retention period, 202
 - using touch to set the retention period, 203
 - viewing WORM file attributes, 205
- wr_throttle
 - mount option, 224
 - samu(1M) command, 314

- write throttle, 224
- writebehind
 - mount option, 221
 - samu(1M) command, 314
- writer mount option, 208
- wrlease
 - mount option, 254
 - samu(1M) command, 317

X

- xntpd(1M) daemon, 244

Z

- zero quotas, 105

