Sun StorEdge™ SAM-FS Troubleshooting Guide

Version 4, Update 4

Sun Microsystems, Inc.
www.sun.com

Part No. 819-2756-10
Sept. 2005, Revision A

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

Contents

Tables

Preface

This manual is written for Sun personnel and customers who need to diagnose and troubleshoot common problems encountered with the installation, configuration, and operation of a Sun StorEdge SAM-FS system. It describes problem situations, the tools needed to diagnose problems, and approaches to problem resolution. The reader is assumed to be familiar with Sun StorEdge SAM-FS and Sun StorEdge QFS installation, configuration, and basic operations.

This manual also describes the steps to prepare for disaster recovery and steps to recover from a disaster, should one occur. It describes the system data (metadata) that you need to protect and how to use that data to reconstruct or recover lost data. The types of data recovery addressed in this manual range from recovering a single lost file to recovering large amounts of data lost in a fire, flood, or other disaster.

Many of the procedures in this manual have to do with a Sun StorEdge QFS file system running in conjunction with a Sun StorEdge SAM-FS storage and archive management system. This combination is known as SAM-QFS. This manual does not provide troubleshooting information for stand-alone Sun StorEdge QFS file systems. For Sun StorEdge QFS information, see the *Sun StorEdge QFS Configuration and Administration Guide*.

Before You Read This Book

You, the system administrator, are assumed to be knowledgeable about Solaris system and network administration procedures, including installation, configuration, creation of accounts, and system backups.

Before you read this book, you need to understand how to administer Sun StorEdge QFS and Sun StorEdge SAM-FS environments as described in the other manuals under "Related Documentation" on page xiii.

How This Book Is Organized

The first three chapters of this book cover general troubleshooting procedures for the Sun StorEdge SAM-FS software and the File System Manager software (previously called SAM-QFS Manager). Chapters 4 through 8 cover file system backup and restoration procedures. Backup procedures described in Chapter 4 are applicable for Sun StorEdge QFS and SAM-QFS file systems and for all types of archive media. The recovery procedures in the remaining chapters apply only to SAM-QFS file systems.

Also, while the procedures in Chapter 5 are for recovering individual files from all supported types of archive media, the recovery procedures for damaged file systems in Chapter 6 apply only to file systems archived on tape or on magneto optical disk. Procedures for recovering file systems archived on hard disks are outside the scope of this manual.

This manual contains the following chapters:

- Chapter 1 provides an overview of the troubleshooting process
- Chapter 2 describes specific troubleshooting features for the Sun StorEdge SAM-FS software
- Chapter 3 describes troubleshooting methods for the File System Manager software
- Chapter 4 describes how to back up data in the Sun StorEdge SAM-FS environment
- Chapter 5 explains how to recover individual data files and directories
- Chapter 6 explains how to recover data from damaged volumes
- Chapter 7 explains how to recover data from damaged file systems
- Chapter 8 provides overall guidelines for recovery after a catastrophic failure.

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

Related Documentation

This manual is part of a set of documents that describe the operations of the Sun StorEdge QFS and Sun StorEdge SAM-FS software products. TABLE P-1 shows the complete documentation set for Version 4, Update 4 (4U4) of these products.

TABLE P-1 Related Documentation

Title	Part Number
Sun StorEdge QFS Installation and Upgrade Guide	819-2757-10
Sun StorEdge QFS Configuration and Administration Guide	819-2758-10
Sun StorEdge SAM-FS File System Configuration and Administration Guide	819-3635-10
Sun StorEdge SAM-FS Installation and Upgrade Guide	819-2754-10
Sun StorEdge SAM-FS Storage and Archive Management Guide	819-2755-10
Sun StorEdge QFS and Sun StorEdge SAM-FS 4.4 Release Notes	818-2759-10

Installation Assistance

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

Accessing Sun Documentation Online

The Sun StorEdge QFS and Sun StorEdge SAM-FS software distribution includes PDF files of the documents for these products. These PDF files can also be accessed online at either of two locations by using one of the two methods below.

- Method 1: Access the documents at `docs.sun.com`.
  1. Go to the following URL:

     `http://docs.sun.com`

     The `docs.sun.com` page displays.
  2. Enter the appropriate product name in the search box.

- Method2: Access the documents at Sun's Network Storage documentation website.
  1. Go to the following URL:

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

     The Storage Software page displays.
  2. Click on the appropriate product link.

Note – Viewing PDF files requires the Acrobat Reader software, which is available for free from the following website: `www.adobe.com`.

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™ Operating System documentation, which is at the following URL:

  `http://docs.sun.com`

Typographic Conventions

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

TABLE P-2 Typographic Conventions

Typeface or Symbol	Meaning	Examples
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output.	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`

`AaBbCc123`	What you type, when contrasted with on-screen computer output.	`%` **`su`**
		`Password:`

AaBbCc123	Book titles; new words or terms; words to be emphasized; and command line variables to be replaced with a real name or value.	Read Chapter 6 in the *User's Guide*. These are called *class* options. You *must* be `root` to do this. To delete a file, type `rm` *filename*.
[ ]	In syntax, brackets indicate that an argument is optional.	`scmadm` [`-d` *sec*] [`-r` *n*[:*n*][,*n*]...] [`-z`]
{ *arg* | *arg*}	In syntax, braces and pipes indicate that one of the arguments must be specified.	`sndradm -b` {*phost* | *shost*}

\	At the end of a command line, the backslash (\) indicates that the command continues on the next line.	`atm90 /dev/md/rdsk/d5 \`
		`/dev/md/rdsk/d1 atm89`

Shell Prompts

TABLE P-3 shows the shell prompts that this manual uses.

TABLE P-3 Shell Prompts

Shell	Prompt
C shell	*machine_name*`%`
C shell superuser	*machine_name*`#`
Bourne shell and Korn shell	`$`
Bourne shell and Korn shell superuser	`#`

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can email your comments to Sun at:

`docfeedback@sun.com`

Please include the part number (819-2756-10) of your document in the subject line of your email.

CHAPTER **1**

Troubleshooting Overview

Sun StorEdge SAM-FS problems are frequently symptoms of incorrect hardware and software configuration during installation or upgrade. This chapter provides basic information on diagnosing and troubleshooting such problems in the Sun StorEdge SAM-FS environment and also discusses preparing a disaster recovery plan.

This chapter includes the following subsections:

Troubleshooting Issues

The following sections provide an overview of some of the hardware and software configuration issues that may be encountered in the Sun StorEdge SAM-FS environment.

The following topics are covered:

Daemons

The following sections describe the daemons that can be present in a SAM-QFS environment and show how to verify the functionality of these daemons.

SAM-QFS Daemons

The process spawner, `init`(1M), starts the `sam-fsd`(1M) daemon based on information defined in `inittab`(4). The `sam-fsd`(1M) daemon provides overall control of the initialization of the SAM-QFS environment. As part of this process, it starts a number of child daemons. These child daemons are as follows:

- The `sam-archiverd`(1M) daemon. Controls the file archiving process in a SAM-QFS environment. The `sam-archiverd`(1M) daemon starts one `sam-arfind`(1M) process per mounted file system. In addition, the `sam-archiverd`(1M) daemon starts a variable number of `sam-arcopy`(1M) processes depending upon the level of archiving activity and number archive requests that are generated by the `sam-arfind`(1M) processes.
- The `sam-stagerd`(1M) daemon. Controls the file staging process. This daemon starts the `sam-stagerd_copy`(1M) processes, which copy archived files from archive media to the online disk cache.
- The `sam-stagealld`(1M) daemon. Controls the associative staging of files.
- The `sam-ftpd`(1M) daemon. Transfers data between local and remote Sun StorEdge SAM-FS systems when Sun SAM-Remote is configured.
- The `sam-amld`(1M) daemon. Initializes several parts of the system and starts the following other daemons as necessary:
  - The `sam-scannerd`(1M) daemon. Monitors all manually mounted removable-media devices. The scanner periodically checks each device for inserted archive media cartridges.
  - The `sam-catserverd`(1M) daemon. Builds and maintains library catalog files for automated libraries.
  - The `sam-robotsd`(1M) daemon. Starts and monitors the robot control daemons for automated libraries and media changers. The `sam-robotsd`(1M) daemon, in turn, starts various daemons depending on the type of robots attached and whether they are direct-attached or network-attached.

Verifying SAM-QFS Daemons

It is possible to determine which daemons and processes should be running for a given configuration based on a knowledge of the SAM-QFS daemons and processes and the circumstances under which they are started. You can check that the expected daemons or processes are running by using the `ps`(1) and `ptree`(1) commands.

CODE EXAMPLE 1-1 assumes that the `ps`(1) command is issued in a SAM-QFS environment that includes a a StorageTek L700 library connected via ACSLS to a SAM-QFS system with two mounted file systems, `samfs1` and `samfs2`. In this example, the `sam-stkd`(1M) daemon is running. This controls the network-attached StorageTek media changers through the ACSAPI interface implemented by the ACSLS software. If such equipment were present, similar daemons would be started for network-attached IBM (`sam-ibm3494d`(1M)) and Sony (`sam-sonyd`(1M)) automated libraries, and for standard direct-attached automated libraries that conform to the SCSI-II standard for media changers (`sam-genericd`(1M)).

CODE EXAMPLE 1-1 Verifying SAM-QFS Daemons

```
skeeball # ps -ef | grep sam-fsd | grep -v grep
    root   656     1  0 10:42:26 ?        0:00 /usr/lib/fs/samfs/sam-fsd
skeeball # ptree 656
656   /usr/lib/fs/samfs/sam-fsd
  681   sam-archiverd
    931   sam-arfind samfs2
    952   sam-arfind samfs1
  683   sam-stagealld
  682   sam-ftpd
  684   sam-stagerd
  685   sam-amld
    687   sam-catserverd 1 2
    689   sam-scannerd 1 2
    690   sam-robotsd 1 2
      691   sam-stkd 1 2 30
        692   /opt/SUNWsamfs/sbin/ssi_so 692 50014 23
        694   sam-stk_helper 1 30
skeeball #
```

▼ To Check `ps`(1) Output and Related Factors

The following steps show you what to look for in the `ps`(1) command's output.

1. **Check the output for missing or duplicate daemon processes and defunct processes.**

   There should be only one of each of these processes, with few exceptions, as follows:

   - There should be one `sam-arfind`(1M) process per mounted file system.
   - There should be one `sam-stkd`, `sam-ibm3494d`, `sam-sonyd` or `sam-genericd` process per automated library defined in the `mcf` file. For more information, see the `sam-robotsd`(1M) man page.
   - There should be zero or more `sam-arcopy`(1M) processes depending on configuration and the archiving load.
   - There should be zero or more `sam-stagerd_copy`(1M) processes depending on configuration and staging load.

2. **Check the configuration files.**

   The `sam-fsd`(1M) daemon reads the following configuration files: `mcf`(4), `defaults.conf`(4), `diskvols.conf`(4), and `samfs.cmd`(4). Verify that these configuration files are error free by issuing the `sam-fsd`(1M) command manually and watching for error messages. As CODE EXAMPLE 1-2 shows, if `sam-fsd`(1M) encounters errors when processing these files, it exits without starting up the SAM-QFS environment.

CODE EXAMPLE 1-2 `sam-fsd`(1M) Output

```
skeeball # sam-fsd
6: /dev/dsk/c1t2d0s0    10   md    samfs1     on     /dev/rdsk/c1t2d0s0
 *** Error in line 6: Equipment ordinal 10 already in use
1 error in '/etc/opt/SUNWsamfs/mcf'
sam-fsd: Read mcf /etc/opt/SUNWsamfs/mcf failed.
skeeball #
```

Configuration Files

After the software packages have been installed, you need to tailor the SAM-QFS configuration files to the site installation in order to bring the system into an operational state. Syntactical and typographical errors in these configuration files manifest themselves in unexpected behavior. TABLE 1-1 shows the relevant files.

TABLE 1-1 Configuration Files and Their Locations

Configuration File Purpose	Default Location
Master configuration file	`/etc/opt/SUNWsamfs/mcf`
`st` device file	`/kernel/drv/st.conf`
`samst`(7) device file	`/kernel/drv/samst.conf`
Device mapping	`/etc/opt/SUNWsamfs/inquiry.conf`
Default settings file	`/etc/opt/SUNWsamfs/defaults.conf`
`mount`(1M) command options	`/etc/opt/SUNWsamfs/samfs.cmd`
Disk archiving configuration file	`/etc/opt/SUNWsamfs/diskvols.conf`
`st.conf` examples file	`/opt/SUNWsamfs/examples/st.conf_changes`
Archiver configuration file	`/etc/opt/SUNWsamfs/archiver.cmd`
Releaser configuration file	`/etc/opt/SUNWsamfs/releaser.cmd`
Stager configuration file	`/etc/opt/SUNWsamfs/stager.cmd`
Recycler configuration file	`/etc/opt/SUNWsamfs/recycler.cmd`

Many of these files are described in the following sections:

- "The `/etc/opt/SUNWsamfs/mcf` File" on page 14
- "The `/kernel/drv/st.conf` File" on page 17
- "The `/kernel/drv/samst.conf` File" on page 18
- "The `/etc/opt/SUNWsamfs/inquiry.conf` File" on page 20
- "The `/etc/opt/SUNWsamfs/defaults.conf` File" on page 21

The later chapters of this manual address the rest of the files introduced in TABLE 1-1.

Log and Trace Files

Using the appropriate log trace files can greatly facilitate the diagnosis of SAM-QFS problems. TABLE 1-2 shows the relevant files.

TABLE 1-2 Log and Trace File Summary

File	Default Location
Sun StorEdge SAM-FS log file	Configurable. Defined in `/etc/syslog.conf`.
System messages file	`/var/adm/messages`.
Device logs	`/var/opt/SUNWsamfs/devlog/`*eq*.
Daemon trace files	Configurable. Defined in `/var/opt/SUNWsamfs/trace`.
Archiver log file	Configurable. Defined in `archiver.cmd`(4).
Releaser log file	Configurable. Defined in `releaser.cmd`(4).
Stager log file	Configurable. Defined in `stager.cmd`(4).
Recycler log file	Configurable. Defined in `recycler.cmd`(4).

The following sections describe how to use the log and trace files when troubleshooting:

- "Enabling System Logging" on page 6
- "Enabling Device Down Notification" on page 7
- "Enabling Daemon Tracing" on page 7
- "Enabling Device Logging" on page 8

Enabling System Logging

The SAM-QFS software makes log entries using the standard Sun StorEdge SAM-FS log file interface (see `syslogd`(1M), `syslog.conf`(4), `syslog`(3C)). All logging is done based on a *level* and a *facility*. The level describes the severity of the reported condition. The facility describes the component of the system sharing information with the `syslogd`(1M) daemon. The SAM-QFS software uses facility `local7` by default.

▼ To Enable System Logging

To enable the `syslogd`(1M) daemon to receive information from the SAM-QFS software for system logging, perform the following steps:

1. **Add a line to the `/etc/syslog.conf` file to enable logging.**

   For example, add a line similar to the following:

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

   You can copy this line from `/opt/SUNWsamfs/examples/syslog.conf_changes`. This entry is all one line, and it has a TAB character (not a space) between the fields.

2. **Use `touch`(1) to create an empty `/var/adm/sam-log` file.**

   For example:

```
skeeball # touch /var/adm/sam-log
```

3. **Send the `syslogd`(1M) process a SIGHUP signal.**

   For example:

CODE EXAMPLE 1-3 Sending a `SIGHUP` to `syslogd`(1M)

```
skeeball # ps -ef | grep syslogd | grep -v grep
    root   216     1  0   Jun 20 ?        0:00 /usr/sbin/syslogd
skeeball # kill -HUP 216
```

4. **Use `vi`(1) or another editor to open the `defaults.conf` file and add the debugging level. (Optional)**

   Perform this step only if you want to increase the logging level.

You can use the `debug` keyword in the `defaults.conf` file to set the default level for the debug flags used by the SAM-QFS daemons for logging system messages. The syntax for this line is as follows:

```
debug = option-list
```

The default debug level is `logging`, so `debug=logging` is the default specification. For *option-list*, specify a space-separated list of debug options. For more information on the options available, see the `samset`(1M) and `defaults.conf`(4) man pages.

Enabling Device Down Notification

The robot daemon, `sam-robotsd`(1M), starts and monitors the execution of the media changer control daemons in SAM-QFS systems. The `sam-amld`(1M) daemon automatically starts the `sam-robotsd`(1M) daemon if there are any media changers defined in the `mcf` file. For more information, see the `sam-robotsd`(1M) man page.

The `sam-robotsd`(1M) daemon executes the `/opt/SUNWsamfs/sbin/dev_down.sh` notification script when any removable media device is marked `down` or `off`. By default, it sends email to `root` with the relevant information. It can be tailored to use `syslogd`(1M) or to interface with the systems management software in use at a site. For more information, see the `dev_down.sh`(4) man page.

Enabling Daemon Tracing

You can enable daemon tracing by configuring settings in the `defaults.conf`(4) file. CODE EXAMPLE 1-4 shows the syntax to use in the `defaults.conf`(4) file.

CODE EXAMPLE 1-4 Syntax to Enable Daemon Tracing for all Daemons

```
trace
all = on
endtrace
```

CODE EXAMPLE 1-4 enables daemon tracing for all daemons. The system writes trace files for each daemon to the following default location:

```
/var/opt/SUNWsamfs/trace/daemon_name
```

Alternatively, trace files can be turned on individually for the `sam-archiverd`(1M), `sam-catserverd`(1M), `sam-fsd`(1M), `sam-ftpd`(1M), `sam-recycler`(1M), and `sam-stagerd`(1M) processes. CODE EXAMPLE 1-5 enables daemon tracing for the archiver in `/var/opt/SUNWsamfs/trace/sam-archiverd`, sets the name of the archiver trace file to *filename*, and defines a list of optional trace events or elements to be included in the trace file as defined in *option-list*.

CODE EXAMPLE 1-5 Syntax to Enable `sam-archiverd`(1M) Tracing

```
trace
sam-archiverd = on
sam-archiverd.file = filename
sam-archiverd.options = option-list
sam-archiverd.size = 10M
endtrace
```

Note that daemon trace files are not automatically rotated by default. As a result, the trace files can become very large, and they might eventually fill the `/var` file system. You can enable automatic trace file rotation in the `defaults.conf`(4) file by using the *daemon-name*`.size` parameter.

The `sam-fsd`(1M) daemon invokes the `trace_rotate.sh`(1M) script when a trace file reaches the specified size. The current trace file is renamed *filename*`.1`; the next newest is renamed *filename*`.2`; and so on, for up to 7 generations. CODE EXAMPLE 1-5 specifies that the archiver trace file should be rotated when its size reaches 10 megabytes.

For detailed information on the events that can be selected, see the `defaults.conf`(4) man page.

Enabling Device Logging

SAM-QFS systems write messages for archiving devices (automated libraries and tape drives) in log files stored in `/var/opt/SUNWsamfs/devlog`. In this directory of files, there is one log file for each device, and these files contain device-specific information. Each removable-media device has its own device log, which is named after its Equipment Ordinal (*eq*) as defined in the `mcf` file. There is also a device log for the Historian (Equipment Type `hy`) with a file name equal to the highest *eq* defined in the `mcf` file incremented by one.

You can use the `devlog` keyword in the `defaults.conf`(4) file to set up device logging using the following syntax:

```
devlog eq [option-list]
```

If *eq* is set to `all`, the event flags specified in *option-list* are set for all devices.

For *option-list*, specify a space-separated list of `devlog` event options. If *option-list* is omitted, the default event options are `err`, `retry`, `syserr`, and `date`. For information on the list of possible event options, see the `samset`(1M) man page.

You can use the `samset`(1M) command to turn on device logging from the command line. Note that the device logs are not maintained by the system, so you must implement a policy at your site to ensure that the log files are routinely rolled over.

CODE EXAMPLE 1-6 shows sample device log output using the default output settings. It shows the first initialization of a 9840A tape drive. The drive is specified as Equipment Ordinal `31` in the `mcf` file.

CODE EXAMPLE 1-6 Device Log Output Example

```
skeeball # cat mcf
#
# Equipment             Eq   Eq    Family   Device   Additional
# Identifier            ORD Type    Set     State    Parameters
#-----------            --- ----   ------   ------   ----------
samfs1                  10   ms    samfs1     on
/dev/dsk/c1t2d0s0       11   md    samfs1     on     /dev/rdsk/c1t2d0s0
#
samfs2                  20   ms    samfs2     on
/dev/dsk/c1t2d0s1       21   md    samfs2     on     /dev/rdsk/c1t2d0s1
#
#
#        ---------- STK ACSLS Tape Library -----------
#
# Equipment                   Eq   Eq  Family Device Additional
# Identifier                  Ord Type  Set   State  Parameters
#-----------                  --- ---- ------ ------ ----------
/etc/opt/SUNWsamfs/stk30       30  sk  stk30    on      -
/dev/rmt/0cbn                  31  sg  stk30    on      -
/dev/rmt/1cbn                  32  sg  stk30    on      -
skeeball #
skeeball # ls /var/opt/SUNWsamfs/devlog
30  31  32  33
skeeball # more /var/opt/SUNWsamfs/devlog/31
2003/06/11 11:33:31*0000 Initialized. tp
2003/06/11 11:33:31*1002 Device is STK     , 9840
2003/06/11 11:33:31*1004 Rev 1.28
2003/06/11 11:33:31*1005 Known as STK 9840 Tape(sg)
2003/06/11 11:33:37 0000 Attached to process 691
2003/06/11 14:31:29 1006 Slot 0
2003/06/11 14:31:29 0000 cdb - 08 00 00 00 50 00
2003/06/11 14:31:29 0000       00 00 00 00 00 00
2003/06/11 14:31:29 0000 sense - f0 00 80 00 00 00 50 12 00 00
2003/06/11 14:31:29 0000         00 00 00 01 00 00 00 00 00 00
```

CODE EXAMPLE 1-6 Device Log Output Example *(Continued)*

```
2003/06/11 14:31:30 0000 cdb - 08 00 00 00 50 00
2003/06/11 14:31:30 0000       00 00 00 00 00 00
2003/06/11 14:31:30 0000 sense - f0 00 80 00 00 00 50 12 00 00
2003/06/11 14:31:30 0000         00 00 00 01 00 00 00 00 00 00
2003/06/11 14:31:31 0000 cdb - 08 00 00 00 50 00
2003/06/11 14:31:31 0000       00 00 00 00 00 00
2003/06/11 14:31:31 0000 sense - f0 00 80 00 00 00 50 12 00 00
2003/06/11 14:31:31 0000         00 00 00 01 00 00 00 00 00 00
2003/06/11 14:31:31 3021 Writing labels
2003/06/11 14:31:32 1006 Slot 0
2003/06/11 14:31:32 3003 Label 700181 2003/06/11 14:31:31 blocksize = 262144
.
.
```

CODE EXAMPLE 1-6 shows a 9840A device being initialized and, some three hours later, a tape from slot 0 being loaded into the tape drive for archiving. The tape is checked three times for its VSN label, and each time the system reports that the media is blank. After three checks, the system concludes that the tape is blank, labels it, and then reports the VSN label (700181), the date, the time, and the media block size.

Troubleshooting Utilities and the `samexplorer`(1M) Diagnostic Report

The SAM-QFS software support several troubleshooting utilities and one diagnostic report, the `samexplorer`(1M) script (called `info.sh`(1M) in versions prior to 4U1). The following sections describe these tools.

Troubleshooting Utilities

TABLE 1-3 lists the utilities that are helpful in diagnosing SAM-QFS configuration problems.

TABLE 1-3 Troubleshooting Utilities

Utility	Description
`sam-fsd`(1M)	Initializes the environment. Debugs basic configuration problems, particularly with new installations.
`samu`(1M)	Full-screen operator interface to SAM-QFS systems. Comprehensive display shows status of file systems and devices. Allows operator to control file systems and removable media devices.

TABLE 1-3 Troubleshooting Utilities *(Continued)*

Utility	Description
`sls`(1)	Sun Microsystems extended version of the GNU `ls`(1M) command. The `-D` option displays extended SAM-QFS attributes.
`samset`(1M)	Sets parameters within the SAM-QFS environment.
`samexplorer`(1M)	Generates SAM-QFS diagnostic reports. Also described in "The `samexplorer`(1M) Script" on page 11.

TABLE 1-3 briefly describes the general form of these utilities. Consult the relevant man pages and the SAM-QFS documentation, particularly *Sun StorEdge QFS Configuration and Administration Guide* and the *Sun StorEdge SAM-FS Storage and Archive Management Guide*, for more information.

The `samexplorer`(1M) Script

The `samexplorer`(1M) script (called `info.sh`(1M) in versions prior to 4U1) collates information from a SAM-QFS environment and writes this to file `/tmp/SAMreport`. The information contained in the SAMreport is an important aid to diagnosing complex SAM-QFS problems, and it is needed by an engineer in the event of an escalation.

The SAMreport includes the following information:

- Packaging, revision levels, and licensing information
- Configuration files (`mcf`(4), `archiver.cmd`(4), `recycler.cmd`(4), `inquiry.conf`(4), `defaults.conf`(4))
- Log files (`sam-log`, messages, `archiver.log`, `recycler.log`, `releaser.log`, trace files)
- Memory dump information

If log files are not routinely collected, an important source of diagnostic information is missing from the SAMreport. It is important to ensure that sites implement a comprehensive logging policy as part of its standard system administration procedures.

It is recommended that the SAMreport be generated in the following circumstances:

- Whenever there is a system panic, core dump, crash, hang, or stall
- As close to any system event as possible

Run the `samexplorer` script and save the SAMreport file before attempting recovery. Ensure that SAMreport is moved from `/tmp` before rebooting. The functionality of `samexplorer` has been fully incorporated into the Sun Explorer

Data Collector, release 4U0. However, `samexplorer` provides a focused set of data tuned to the SAM-QFS environment that can be quickly and simply collected and sent to escalation engineers for rapid diagnosis.

Common Problems

The following sections describe various system configuration problems that can be diagnosed and remedied:

- "Hardware Configuration Problems" on page 12
- "SAN-attached Device Configuration Problems" on page 13

Hardware Configuration Problems

SAM-QFS problems can turn out to be hardware related. Before embarking on an extensive troubleshooting exercise, ascertain the following:

- The system hardware is correctly set up and visible to the SAM-QFS system.
- The devices have been identified and configured correctly for SAM-QFS operations.

▼ To Verify Hardware

It is easiest to verify the hardware configuration by performing the following procedure. However, this procedure requires you to shut down the system. If the system cannot be shut down, consult the `/var/adm/messages` file for the device check-in messages from the last reboot.

To verify that the Solaris OS can communicate with the devices attached to the server, perform the following steps:

1. **Shut down the system.**
2. **Issue the** `probe-scsi-all` **command at the** `ok` **prompt.**
3. **Monitor the boot-up sequence messages.**

   While monitoring the messages, identify the check-in of the expected devices.

CODE EXAMPLE 1-7 shows the `st` tape devices checking in.

CODE EXAMPLE 1-7 Check In of `st` Tape Devices

```
Jun  9 13:29:39 skeeball scsi: [ID 365881 kern.info]
/pci@1f,0/pci@1/scsi@3/st@4,0 (st18):
Jun  9 13:29:39 skeeball     <StorageTek 9840>
Jun  9 13:29:39 skeeball scsi: [ID 193665 kern.info] st18 at glm2: target 4 lun 0
Jun  9 13:29:39 skeeball genunix: [ID 936769 kern.info] st18 is
/pci@1f,0/pci@1/scsi@3/st@4,0
Jun  9 13:29:39 skeeball scsi: [ID 365881 kern.info]
/pci@1f,0/pci@1/scsi@3/st@5,0 (st19):
Jun  9 13:29:39 skeeball     <StorageTek 9840>
Jun  9 13:29:39 skeeball scsi: [ID 193665 kern.info] st19 at glm2: target 5 lun 0
Jun  9 13:29:39 skeeball genunix: [ID 936769 kern.info] st19 is
/pci@1f,0/pci@1/scsi@3/st@5,0.
.
```

If devices do not respond, consult your Solaris documentation for information on configuring the devices for the Solaris OS.

If you have verified that the hardware has been installed and configured correctly and that no hardware faults are present, the next step in diagnosing an installation or configuration problem is to check that the expected SAM-QFS daemons are running. For more information on the daemons, see "Daemons" on page 2.

SAN-attached Device Configuration Problems

SAN-attached devices, such as fibre channel drives and automated libraries, should be checked to ensure that they are configured and that they are visible to the Solaris OS through the `cfgadm`(1M) command. CODE EXAMPLE 1-8 illustrates this for a fabric-attached library controller and drives.

CODE EXAMPLE 1-8 `cfgadm`(1M) Command Output

```
# cfgadm -al
Ap_Id                  Type            Receptacle      Occupant        Condition
n
c0                      scsi-bus        connected        configured     unknown
c0::dsk/c0t0d0          disk            connected        configured     unknown
c0::dsk/c0t6d0          CD-ROM          connected        configured     unknown
c1                      scsi-bus        connected        configured     unknown
c2                      scsi-bus        connected        unconfigured   unknown
c4                    fc-fabric        connected        configured      unknown
c4::210000e08b0645c1  unknown         connected       unconfigured    unknown
.
.
```

CODE EXAMPLE 1-8 `cfgadm`(1M) Command Output *(Continued)*

```
c4::500104f00041182b    med-changer     connected       configured      unknown
c4::500104f00043abfc    tape            connected       configured      unknown
c4::500104f00045eeaf    tape            connected       configured      unknown
c4::5005076300416303    tape            connected       configured      unknown
.
```

If devices are in an unconfigured state, use the `cfgadm`(1M) command with its `-c` *configure* option to configure the devices into the Solaris environment. It is important to understand the SAN configuration rules for Fibre Channel tape devices and libraries. For more information, see the latest Sun StorEdge Open SAN Architecture or the SAN Foundation Kit Package documentation for more information.

Troubleshooting Configuration Files

This section describes specific troubleshooting procedures for identifying issues with the Sun StorEdge SAM-FS and Sun StorEdge QFS configuration files.

The `/etc/opt/SUNWsamfs/mcf` File

The `mcf`(4) file defines the SAM-QFS devices and device family sets.

The `mcf` file is read when `sam-fsd`(1M) is started. It can be changed at any time, even while `sam-fsd` is running, but `sam-fsd`(1M) recognizes `mcf` file changes only when the daemon is restarted. CODE EXAMPLE 1-9 shows an `mcf` file for a SAM-QFS environment.

CODE EXAMPLE 1-9 Example SAM-QFS `mcf` File

```
#
# SAM-QFS file system configuration example
#
# Equipment       Eq Eq Family Dev Additional
# Identifier      Or Tp Set    St  Parameters
# --------------- -- -- ------ --- ----------
samfs1            60 ms samfs1
/dev/dsk/c1t1d0s6 61 md samfs1 on
/dev/dsk/c2t1d0s6 62 md samfs1 on
/dev/dsk/c3t1d0s6 63 md samfs1 on
/dev/dsk/c4t1d0s6 64 md samfs1 on
/dev/dsk/c5t1d0s6 65 md samfs1 on
```

CODE EXAMPLE 1-9 Example SAM-QFS `mcf` File *(Continued)*

```
#
samfs2             2 ms samfs2
/dev/dsk/c1t1d0s0 15 md samfs2 on
/dev/dsk/c1t0d0s1 16 md samfs2 on
#
/dev/samst/c0t2d0 20 od -      on
#
/dev/samst/c1t2u0 30 rb hp30   on   /var/opt/SUNWsamfs/catalog/hp30_cat
/dev/samst/c1t5u0 31 od hp30   on
/dev/samst/c1t6u0 32 od hp30   on
#
/dev/rmt/0cbn     40 od -      on
#
/dev/samst/c1t3u1 50 rb ml50   on   /var/opt/SUNWsamfs/catalog/ml50_cat
/dev/rmt/2cbn     51 tp ml50   on
```

The *Sun StorEdge QFS Configuration and Administration Guide* describes the format of the `mcf` file in detail.

The most common problems with the `mcf` file are syntactical and typographical errors. The `sam-fsd`(1M) command is a useful tool in debugging the `mcf` file. If an error is encountered by `sam-fsd`(1M) as it processes the `mcf` file, it writes error messages to the Sun StorEdge SAM-FS log file (if configured). Errors detected in the following other files, if present, are also reported:

- `diskvols.conf`
- `samfs.cmd`
- `defaults.conf`

For a newly created or modified `mcf` file, run the `sam-fsd`(1M) command and check for error messages. If necessary, correct the `mcf` file and rerun the `sam-fsd`(1M) command to ensure that the errors have been corrected. Repeat this process until all errors have been eliminated. When the `mcf` file is error free, reinitialize the `sam-fsd`(1M) daemon by sending it the `SIGHUP` command. CODE EXAMPLE 1-10 shows this process.

CODE EXAMPLE 1-10 Checking the `mcf` File

```
skeeball # sam-fsd
6: /dev/dsk/c1t2d0s0    10   md    samfs1     on     /dev/rdsk/c1t2d0s0
 *** Error in line 6: Equipment ordinal 10 already in use
1 error in '/etc/opt/SUNWsamfs/mcf'
sam-fsd: Read mcf /etc/opt/SUNWsamfs/mcf failed.
skeeball #
skeeball # cat mcf
#
# Equipment             Eq   Eq    Family   Device   Additional
```

CODE EXAMPLE 1-10 Checking the `mcf` File *(Continued)*

```
# Identifier            ORD Type    Set     State   Parameters
#-----------            --- ----   ------   ------   ----------
samfs1                  10   ms    samfs1     on
/dev/dsk/c1t2d0s0       10   md    samfs1     on
#
samfs2                  20   ms    samfs2     on
/dev/dsk/c1t2d0s1       21   md    samfs2     on
#
#
#        ---------- STK ACSLS Tape Library -----------
#
# Equipment                   Eq   Eq  Family Device Additional
# Identifier                  Ord Type  Set   State  Parameters
#-----------                  --- ---- ------ ------ ----------
/etc/opt/SUNWsamfs/stk30       30  sk  stk30    on
/dev/rmt/0cbn                  31  sg  stk30    on
/dev/rmt/1cbn                  32  sg  stk30    on
skeeball #
<correct error>
skeeball #
skeeball # sam-fsd
Trace file controls:
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-ftpd      /var/opt/SUNWsamfs/trace/sam-ftpd
              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
Would stop sam-archiverd()
Would stop sam-ftpd()
Would stop sam-stagealld()
Would stop sam-stagerd()
```

CODE EXAMPLE 1-10 Checking the `mcf` File *(Continued)*

```
Would stop sam-amld()
skeeball #
skeeball # samd config
skeeball #
```

Enable the changes to the `mcf` file for a running system by running the `samd`(1M) command with its `config` option (as shown in CODE EXAMPLE 1-10) or by sending the `SIGHUP` signal to `sam-fsd`(1M). Note that the procedure for reinitializing `sam-fsd`(1M) to make it recognize `mcf` file modifications varies depending on the nature of the changes implemented in the `mcf` file. For more information, see the *Sun StorEdge QFS Configuration and Administration Guide* for the procedures to be followed in specific circumstances.

Verifying `mcf` Drive Order Matching for Direct-attached Libraries

For libraries with more than a single drive, the order in which drive entries appear in the `mcf` file must match the order in which they are identified by the library controller. The drive that the library controller identifies as the first drive must be the first drive entry for that library in the `mcf`, and so on. To check the drive order for a direct-attached library, follow the instructions in the "Checking the Drive Order" section of the *Sun StorEdge SAM-FS Installation and Upgrade Guide*.

Verifying `mcf` Drive Order Matching for Network-attached Libraries

Network-attached libraries use different procedures from direct-attached. The difference is due to the fact that drive order for a network-attached library is defined by the library control software.

For example, for a network-attached StorageTek library, the drive mapping in the ACSLS parameters file must match the drives as presented by the ACSLS interface. In this case, the procedure is similar to that for a library without a front panel, except that an additional check is necessary to ensure that the ACSLS parameters file mapping is correct.

The `/kernel/drv/st.conf` File

Some tape devices that are compatible with SAM-QFS software are not supported by default in the Solaris operating system (OS) kernel. The file `/kernel/drv/st.conf` is the Solaris `st`(7D) tape driver configuration file for all supported tape drives. The file can be modified to enable operation of normally unsupported drives to work with SAM-QFS system. Attempting to use any such device in the SAM-QFS environment without updating the `st.conf` file, or with an incorrectly-modified file, causes the system to write messages such as the following to device log file:

```
Aug  3  19:43:36 samfs2 scanner[242]: Tape device 92 is default
type. Update /kernel/drv/st.conf
```

If your configuration is to include devices not supported by the Solaris OS, consult the following file for instructions on how to modify the `st.conf` file:

```
/opt/SUNWsamfs/examples/st.conf_changes
```

For example, the IBM LTO drive is not supported by default in Solaris kernel. CODE EXAMPLE 1-11 shows the lines you need to add to the `st.conf` file in order to include IBM LTO drives in a SAM-QFS environment.

CODE EXAMPLE 1-11 Lines to be Added to `st.conf`

```
 "IBM    ULTRIUM-TD1",           "IBM Ultrium",  "CLASS_3580",
CLASS_3580      =       1,0x24,0,0x418679,2,0x00,0x01,0;
```

The `st.conf` file is read only when the `st` driver is loaded, so if the `/kernel/drv/st.conf` file is modified, perform one of the following actions in order to direct the system to recognize the changes:

- Use the `unload`(1M) and `modload`(1M) command to reload the driver.
- Reboot the system.

The `/kernel/drv/samst.conf` File

The `samst`(7) driver for SCSI media changers and optical drives is used for direct-attached SCSI or Fibre Channel tape libraries and for magneto-optical drives and libraries.

As part of the installation process, the SAM-QFS software creates entries in the `/dev/samst` directory for all devices that were attached and recognized by the system before the `pkgadd`(1M) command was entered to begin the installation.

If you add devices after running the `pkgadd`(1M) command, you must use the `devfsadm`(1M) command, as follows, to create the appropriate device entries in `/dev/samst`:

```
# /usr/sbin/devfsadm -i samst
```

After the command is issued, verify that the device entries have been created in `/dev/samst`. If they have not, then perform a reconfiguration reboot and attempt to create the entries again.

If the `/dev/samst` device is not present for the automated library controller, the `samst.conf` file might need to be updated. In general, Fibre Channel libraries, libraries with targets greater than 7, and libraries with LUNs greater than 0 require the `samst.conf` file to be updated. To add support for such libraries, add a line similar to the following to the `/kernel/drv/samst.conf` file:

```
name="samst" parent="fp" lun=0 fc-port-wwn="500104f00041182b";
```

In the previous example line, `500104f00041182b` is the WWN port number of the Fibre-attached automated library. If you need to, you can obtain the WWN port number from the `cfgadm`(1M) command's output. CODE EXAMPLE 1-12 shows this command.

CODE EXAMPLE 1-12 Using `cfgadm`(1M) to obtain the WWN

```
# cfgadm -al
Ap_Id           Type            Receptacle      Occupant        Condition
c0                      scsi-bus        connected       configured      unknown
c0::dsk/c0t0d0          disk            connected       configured      unknown
c0::dsk/c0t6d0          CD-ROM          connected       configured      unknown
c1                      scsi-bus        connected       configured      unknown
c2                      scsi-bus        connected       unconfigured    unknown
c4                      fc-fabric       connected       configured      unknown
c4::210000e08b0645c1    unknown      connected       unconfigured     unknown
.
.
c4::500104f00041182b    med-changer     connected       configured      unknown
c4::500104f00043abfc    tape            connected       configured      unknown
c4::500104f00045eeaf    tape            connected       configured      unknown
c4::5005076300416303    tape            connected       configured      unknown
.
```

For network-attached tape libraries such as a StorageTek library controlled by ACSLS, the `samst` driver is not used, and no `/dev/samst` device entries are created.

The `/etc/opt/SUNWsamfs/inquiry.conf` File

The `/etc/opt/SUNWsamfs/inquiry.conf` file defines vendor and product identification strings for recognized SCSI or fibre devices and matches these with SAM-QFS product strings. If you have devices that are not defined in `inquiry.conf`, you need to update the file with the appropriate device entries. This is not a common practice because the great majority of devices are defined in the file. CODE EXAMPLE 1-13 shows an fragment of the `inquiry.conf` file.

CODE EXAMPLE 1-13 Part of the `inquiry.conf` File

```
"ATL",          "ACL2640",      "acl2640"       # ACL 2640 tape library
"HP",           "C1160A",       "hpoplib"       # HP optical library
"IBM"           "03590",        "ibm3590"       # IBM3590 Tape
"MTNGATE"       "V-48"          "metd28"        # metrum v-48 tape library
"OVERLAND",     "LXB",          "ex210"     # Overland LXB2210 robot
"Quantum"       "DLT2000",      "dlt2000"       # digital linear tape
"STK",          "9490",      "stk9490"       # STK 9490 tape drive
"STK",          "97",           "stk97xx"       # STK 9700 series SCSI
"STK",          "SD-3"          "stkd3"     # STK D3 tape drive
```

If changes to this file are required, you must make them and then reinitialize your SAM-QFS software by issuing the following commands:

CODE EXAMPLE 1-14 Reinitializing the SAM-QFS Software

```
# samd stop
# samd config
```

If the system detects errors in the `inquiry.conf` file during reinitialization, it writes messages to the Sun StorEdge SAM-FS log file. Check for error messages similar to those shown in CODE EXAMPLE 1-15 after making changes to `inquiry.conf` and reinitializing the SAM-QFS software.

CODE EXAMPLE 1-15 Messages Regarding `inquiry.conf` Problems

```
.
May 22 16:11:49 ultra1 samfs[15517]: Unknown device, eq 30
("/dev/samst/c0t2u0"), dtype (0x8)
May 22 16:11:49 ultra1 samfs[15517]: Vender/product OVERLAND LXB.
May 22 16:11:49 ultra1 samfs[15517]: Update /etc/opt/SUNWsamfs/inquiry.conf (see
inquiry.conf(4)).
May 22 16:11:49 ultra1 samfs[15517]: Device being offed eq 30.
.
```

The `/etc/opt/SUNWsamfs/defaults.conf` File

The `defaults.conf` configuration file allows you to establish certain default parameter values for a SAM-QFS environment. The system reads the `defaults.conf` file is when `sam-fsd`(1M) is started or reconfigured. It can be changed at any time while the `sam-fsd`(1M) daemon is running. The changes take effect when the `sam-fsd`(1M) daemon is restarted, or when it is sent the signal `SIGHUP`. Temporary changes to many values can be made using the `samset`(1M) command.

The `sam-fsd`(1M) command is also useful for debugging the `defaults.conf`(4) file. If the `sam-fsd`(1M) daemon encounters an error as it processes the `defaults.conf`(4) file, it writes error messages to the Sun StorEdge SAM-FS log file.

For a newly created or modified `defaults.conf`(4) file, run the `sam-fsd`(1M) command and check for error messages. If necessary, correct the file and rerun the `sam-fsd`(1M) command to ensure that the errors have been corrected. Repeat this process until all errors have been eliminated.

If you modify the `defaults.conf`(4) file on a running system, you need to reinitialize it by restarting the `sam-fsd`(1M) daemon. You can use the `samd`(1M) command with its `config` option to restart `sam-fsd`(1M). See the *Sun StorEdge QFS Configuration and Administration Guide* for the procedures to be followed in specific circumstances.

Planning for Disaster Recovery

Data must be backed up and disaster recovery processes must be put in place so that data can be retrieved if any of the following occur:

- Data is accidentally deleted
- Storage media fail
- Systems fail
- Any combination of the above events occurs on a small or large scale

Chapter 4 provides the information you need to know about backing up metadata and other important configuration data. The remaining chapters in this manual describe how to use the data you back up to recover from various types of disasters.

Setting up processes for doing backups and system dumps is only part of preparing to recover from a disaster. The following tasks are also necessary:

- Documenting everything
  - Document your hardware configuration, backup policies and scripts and all of your restoration processes.
  - Keep paper copies of the documents off-site with copies of the backup media.
- Verifying that the files and the system are actually recoverable
  - Test all scripts that you create (see "Testing Backup Scripts and `cron` Jobs" on page 23).
  - Routinely test the retrieval procedures that are described in the other chapters in this manual. See "Testing Backup and Recovery Methods" on page 22.

Recovering From Failure of the Operating Environment Disk

When a disk containing the operating environment for a system fails, after you replace the defective disk(s), you need to do what is called *bare metal recovery* before you can do anything else. Two bare metal recovery approaches are available:

- Reinstall the operating environment, patches, and backed-up configuration files

  This process is slower than the second alternative described below.
- Restore a system image backup made ahead of time on a separate hard disk.

  Image backups need to be made only when system configuration changes are made. The downside to this approach is that it is difficult to safely transport hard disks to off site storage.

Testing Backup and Recovery Methods

After you have done all the recovery preparations described in this chapter, do the test described in the following sections:

- "Testing Backup Scripts and `cron` Jobs" on page 23
- "Testing the Disaster Recovery Process" on page 23

Testing Backup Scripts and `cron` Jobs

Always test backup scripts and `cron`(1) jobs on a development or test system before rolling it out to all systems.

- Test each script's syntax.
- Test each script on one system.
- Test each script on a small number of systems.
- Try to simulate every possible error a script might encounter in the middle of the backup:
  - Eject the volume.
  - Switch the machine off.
  - Pull out the network connection.
  - Switch off the backup server or device.

Testing the Disaster Recovery Process

Use the information in the other chapters in this manual to do the following tests, to verify how well your disaster recovery process works. Do these tests periodically. Especially make it a point to do these tests anytime you make changes to the software.

- Restore a single file that is currently on the system.
- Restore an older version of a file.
- Restore an entire file system and compare it against the original.
- Enact a scenario where the system is down and restore the system.
- Retrieve some volumes from off-site storage.
- Enact a scenario in which last night's backup failed, and you need to restore data using system and archiver logs.
- Enact a scenario in which the system is destroyed and recover the system's data.
- Enact a scenario in which the disk containing the operating environment fails.

CHAPTER 2

Troubleshooting Sun StorEdge SAM-FS Software

This chapter describes how to troubleshoot basic Sun StorEdge SAM-FS functions. It covers the following topics:

- "Troubleshooting the Archiver" on page 25
- "Troubleshooting the Releaser" on page 28
- "Troubleshooting the Recycler" on page 29

Troubleshooting the Archiver

The archiver automatically writes SAM-QFS files to archive media. Operator intervention is not required to archive and stage the files. The archiver starts automatically when a SAM-QFS file system is mounted. You can customize the archiver's operations for your site by inserting archiving directives into the following file:

`/etc/opt/SUNWsamfs/archiver.cmd`

Upon initial setup, the archiver might not perform the tasks as intended. Make sure that you are using the following tools to monitor the archiving activity of the system:

- The File System Manager software. To display archiving activity go to the Servers page and click the name of the server for which you want to display archiving activity. Click the Jobs tab to display the Current Jobs Summary page. Choose whether you want to display current, pending, or all archiving activity by clicking the appropriate local tab under the Jobs tab. From the Filter menu, choose Archive Copy or Archive Scan to view all jobs of either type.

  For complete information on using the File System Manager to monitor jobs, see the File System Manager online help file.

- `samu`(1M) utility's `a` display. This display shows archiver activity for each file system. It also displays archiver errors and warning messages, such as the following:

```
Errors in archiver commands - no archiving will be done
```

  The `samu`(1M) utility's `a` display includes messages for each file system. It indicates when the archiver will scan the `.inodes` file again and the files currently being archived.

- Archive logs. You can define these logs in the `archiver.cmd` file, and you should monitor them regularly to ensure that files are archived to volumes. Archive logs can become excessively large and should be reduced regularly either manually or by using a `cron`(1) job. Archive these log files for safekeeping because the information enables data recovery.
- `sfind`(1). Use this command to check periodically for unarchived files. If you have unarchived files, make sure you know why they are not being archived.
- `sls`(1). Files are not considered for release unless a valid archive copy exists. The `sls -D` command displays inode information for a file, including copy information.

Note – Output from the `sls -D` command might show the word `archdone` on a file. This is not an indication that the file has an archive copy. It is only an indication that the file has been scanned by the archiver and that all the work associated with the archiver itself has been completed. An archive copy exists only when you can view the copy information displayed by the `sls`(1) command.

Occasionally, you might see messages to indicate that the archiver either has run out of space on cartridges or has no cartridges. These messages are as follows:

- When the archiver has no cartridges assigned to an archive set, it issues the following message:

```
No volumes available for Archive Set setname
```

- When the archiver has no space on the cartridges assigned to an archive set, it issues the following message:

```
No space available on Archive Set setname
```

Why Files Are Not Archiving

The following checklist includes reasons why your Sun StorEdge SAM-FS environment might not be archiving files.

- The `archiver.cmd` file has a syntax error. Run the `archiver -lv` command to identify the error, then correct the flagged lines.
- The `archiver.cmd` file has a `wait` directive in it. Either remove the `wait` directive or override it by using the `samu`(1M) utility's :`arrun` command.
- No volumes are available. You can view this from `archiver`(1M) `-lv` command output. Add more volumes as needed. You might have to export existing cartridges to free up slots in the automated library.
- The volumes for an archive set are full. You can export cartridges and replace them with new cartridges (make sure that the new cartridges are labeled), or you can recycle the cartridges. For more information on recycling, see "Recycling" on page 147.
- The VSN section of the `archiver.cmd` file fails to list correct media. Check your regular expressions and VSN pools to ensure that they are correctly defined.
- There is not enough space to archive any file on the available volumes. If you have larger files and it appears that the volumes are nearly full, the cartridges might be as full as the Sun StorEdge SAM-FS environment allows. If this is the case, add cartridges or recycle.

  If you have specified the `-join path` parameter, and there is not enough space to archive all the files in the directory to any volume, no archiving occurs. You should add cartridges, recycle, or use one of the following parameters: `-sort path` or `-rsort path`.
- The `archiver.cmd` file has the `no_archive` directive set for directories or file systems that contain large files.
- The `archive`(1) `-n` (archive never) command has been used to specify too many directories, and the files are never archived.
- Large files are busy. Thus, they never reach their archive age and are not archived.
- Hardware or configuration problems exist with the automated library.
- Network connection problems exist between client and server. Ensure that the client and the server have established communications.

Additional Archiver Diagnostics

In addition to examining the items on the previous list, you should check the following when troubleshooting the archiver.

- The syslog file (by default, `/var/adm/sam-log`). This file can contain archiver messages that can indicate the source of a problem.
- Volume capacity. Ensure that all required volumes are available and have sufficient space on them for archiving.
- If the archiver appears to cause excessive, unexplainable cartridge activity or appears to be doing nothing, turn on the trace facility and examine the trace file. For information on trace files, see the `defaults.conf`(4) man page.
- You can use the `truss`(1) `-p` *pid* command on the archiver process (`sam-archiverd`) to determine the system call that is not responding. For more information on the `truss`(1) command, see the `truss`(1) man page.
- The `showqueue`(1M) command displays the content of the archiver queue files. You can use this command to observe the state of archiver requests that are being scheduled or archived. Any archive request that cannot be scheduled generates a message that indicates the reason. This command also displays the progress of archiving.

Why Files Are Not Releasing

The archiver and the releaser work together to balance the amount of data available on the disk cache. The main reason that files are not released automatically from disk cache is that they have not yet been archived.

For more information on why files are not being released, see the following section.

Troubleshooting the Releaser

There can be several reasons for the releaser to not release a file. Some possible reasons are as follows:

- Files can be released only after they are archived. There might not be an archive copy. For more information about this, see “Why Files Are Not Archiving” on page 27.
- The archiver requested that a file not be released. This can occur under the following conditions:
    - The archiver has just staged an offline file to make an additional copy.
    - The `-norelease` directive in the `archiver.cmd` file was set and all the copies flagged `-norelease` have not been archived. Note that the releaser summary output displays the total number of files with the `archnodrop` flag set.

- The file is set for partial release, and the file size is less than or equal to the partial size rounded up to the disk allocation unit (DAU) size (block size).
- The file changed residence in the last *min_residence_age* minutes.
- The `release -n` command has been used to prevent directories and files from being released.
- The `archiver.cmd` file has the `-release n` option set for too many directories and files.
- The releaser high watermark is set too high, and automatic releasing occurs too late. Verify this in the `samu`(1M) utility's `m` display or with File System Manager, and lower this value.
- The releaser low watermark is set too high, and automatic releasing stops too soon. Check this in the `samu`(1M) utility's `m` display, or with File System Manager, and lower this value.
- Large files are busy. They will never reach their archive age, never be archived, and never be released.

Troubleshooting the Recycler

The most frequent problem encountered with the recycler occurs when the recycler generates a message similar to the following when it is invoked:

```
Waiting for VSN mo:OPT000 to drain, it still has 123 active archive
copies.
```

One of the following conditions can cause the recycler to generate this message:

- Condition 1: The archiver fails to rearchive the 123 archive copies on the volume.
- Condition 2: The 123 archive copies do not refer to files in the file system. Rather, they refer to 123 metadata archive copies.

Condition 1 can exist for one of the following reasons:

- Files that need to be rearchived are marked `no_archive`.
- Files that need to be rearchived are in the `no_archive` archive set.
- Files cannot be archived because there are no available VSNs.
- The `archiver.cmd` file contains a `wait` directive.

To determine which condition is in effect, run the recycler with the `-v` option. As CODE EXAMPLE 2-1 shows, this option displays the path names of the files associated with the 123 archive copies in the recycler log file.

CODE EXAMPLE 2-1 Recycler Messages

```
Archive copy 2 of /sam/fast/testA resides on VSN LSDAT1
Archive copy 1 of /sam3/tmp/dir2/filex resides on VSN LSDAT1
Archive copy 1 of Cannot find pathname for file system /sam3
inum/gen 30/1 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gunk/tstfilA00 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gunk/tstfilF82 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gunk/tstfilV03 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gink/tstfilA06 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gink/tstfilA33 resides on VSN LSDAT1
Waiting for VSN dt:LSDAT1 to drain, it still has 8 active archive
copies.
```

In this example output, messages containing seven path names are displayed along with one message that includes `Cannot find pathname...` text. To correct the problem with `LSDAT1` not draining, you need to determine why the seven files cannot be rearchived. After the seven files are rearchived, only one archive copy is not associated with a file. Note that this condition should occur only as the result of a system crash that partially corrupted the `.inodes` file.

To solve the problem of finding the path name, run `samfsck`(1M) to reclaim orphan inodes. If you choose not to run `samfsck`(1M), or if you are unable to unmount the file system to run `samfsck`(1M), you can manually relabel the cartridge after verifying that the `recycler -v` output is clean of valid archive copies. However, because the recycler continues to encounter the invalid inode remaining in the `.inodes` file, the same problem might recur the next time the VSN is a recycle candidate.

Another recycler problem occurs when the recycler fails to select any VSNs for recycling. To determine why each VSN was rejected, you can run the recycler with the `-d` option. This displays information on how the recycler selects VSNs for recycling.

CHAPTER **3**

Troubleshooting the File System Manager

This chapter describes how to troubleshoot issues that might arise when using the File System Manger software.

This chapter covers the following topics:

- "File System Manager Messages" on page 31
- "Log and Trace Files" on page 33
- "Remote Procedure Call (RPC) Daemon Information" on page 37

File System Manager Messages

This section shows some of the messages you might see when using the File System Manager software.

Message:

```
An unrecoverable error occurred during the page display.
If the problem persists, please restart the web server.
```

Action:

Click the HOME button to return to the Server Selection page, which is the default page of the File System Manager application.

If the system cannot display the Server Selection page, go to the web server and enter the following command to restart the web server:

```
# /usr/sbin/smcwebserver restart
```

Contact your Sun support representative if the problem persists.

Message:

```
HTTP 500 Internal server error
```

Action:

Go to the web server and run the following command to restart the web server:

```
# /usr/sbin/smcwebserver restart
```

Contact your Sun support representative if the problem persists.

Message:

```
The page cannot be displayed.
```

Action:

Go to the web server and run the following command to restart the web server:

```
# /usr/sbin/smcwebserver restart
```

Contact your Sun support representative if the problem persists.

Message:

```
Starting Sun(TM) Web Console Version 2.2.4.
Startup failed. See /var/log/webconsole/console_debug_log for
detailed error information.
```

Action:

Examine the contents of the following file on the web server:

```
/var/log/webconsole/console_debug_log
```

If the log says the port(6789) is in use by some other process, type commands shown in CODE EXAMPLE 0-1.

CODE EXAMPLE 0-1 Restarting the Web Console

```
# pkill -9 noaccess
# /usr/sbin/smcwebserver restart
```

Contact your Sun support representative if the problem persists.

Message:

```
Failed to create the filesystem
mount_samfs: fopen(mnttab) error: : Too many open files
```

Action:

The system generates this message if you are trying to create a file system with a large number of LUNs. To remedy this problem, use the following procedure:

1. **On the file system server, use the** `ps`**(1) and** `grep`**(1) commands to find the process ID for the** `fsmgmtd` **process.**

   For example:

   ```
   # ps -ef | grep fsmgmtd
   ```

2. **Use the** `plimit`**(1) command to increase the descriptors for the process.**

   For example:

   ```
   # plimit -n 512 process-id
   ```

   For *process-id*, specify the process number.

3. **Create the file system.**

Log and Trace Files

The File System Manager software automatically enables logging when it is installed, but if you want tracing, you have to enable it manually. To enable tracing for File System Manager, use the instructions in “Tracing” on page 35.

Log rotation is not supported for log or trace files.

TABLE 3-1 lists the files that the File System Manager uses for logging and tracing.

TABLE 3-1 File System Manager Log and Trace Files

Activity	File Location	Created by the User?
File System Manager Logging	`/var/log/webconsole/fsmgr.log`	No
TomCat Web Console Logging	`/var/log/webconsole/console_debug_log`	No
Tracing for File System Manager and native code	`/var/log/webconsole/fsmgr.trace_syslog`	Yes

The following sections describe the log and trace files.

File System Manager Logging

The File System Manager software creates the `fsmgr.log` log file when the application starts. It records information about operations that the user performs, and whether those operations were successful. Do not delete or modify this file. If you delete or modify this file, logging stops. When the web server restarts, it erases the contents of this file and creates a new `fsmgr.log` file.

The File System Manager software uses an additional file, `/var/webconsole/fsmgr.log.lck`, to ensure that only one process at a time writes to the log file. Do not delete or modify this lock file.

Web Server Logging

The Sun Common Console Framework creates the `/var/webconsole/console_debug_log` file. It includes console-specific information such as environment variable setting that the console uses and a record of users logged in to the console.

If this file becomes too large, you can delete this file. The system creates another instance of this file the next time the web server restarts.

File System Manager Portal Agent Configuration and Log Files

When the File System Manager software is installed, the File System Manager Portal Agent is also installed. This application acts as a information source for the Sun StorEdge Management Portal application. By default, the File System Manager Portal Agent is disabled. It should only be enabled if you are using the Sun StorEdge Management Portal software. The agent uses the same underlying software as the File System Manager. The following files are used for configuring and logging data from the File System Manager Portal Agent.

- `/etc/opt/SUNWfsmgr/agent/conf.sh` - The configuration script that is used when starting the Tomcat process. It defines location of TomCat, Java and other critical components.
- `/var/opt/SUNWfsmgr/agent/tomcat/logs` - Contains the following log files.
  - `catalina.out` - This is the general log file. It contains the log message output from both Tomcat and the agent servlet. Any errors will cause log messages to be written to this file.
  - `fsmgr.`*<date-stamp>*`.log` - This is the application and servlet log file. It contains messages specific to the loading and the running of the agent servlet. It also contains stack trace and fatal error information from the underlying software.

To verify that the agent is running, check the `catalina.out` log file or use the `ps` and `grep` commands to find the agent process.

```
# /usr/ucb/ps -augxww | grep SUNWfsmgr/agent/tomcat
```

Tracing

The File System Manager trace file records the following information:

- Messages regarding whether operations were successful.
- Functions invoked with the application stack. This can be verbose.
- Messages that are important to developers for debugging purposes.

Tracing is not enabled by default.

▼ To Enable Tracing for File System Manager and Native Code

The `syslog` daemon performs detailed tracing for File System Manager and for native code. Use the following procedure to enable detailed tracing for File System Manager and for native code.

1. **Use the `touch`(1) command to create the trace file.**

   For example:

```
# touch /var/log/webconsole/fsmgr.trace_syslog
```

2. **Use `vi`(1) or another editor to add the following line to the `/etc/syslog.conf` file:**

```
local6.debug   /var/log/webconsole/fsmgr.trace_syslog
```

   Use a tab character to separate the two fields in this line.

3. **Type the following command:**

```
# pkill -HUP syslogd
```

4. **(Optional) Enable trace file rotation.**

   Trace files can become very large. Use `logadm`(1M) to manage the trace file for File System Manager.

Note – You cannot use the `log_rotate.sh`(1M) script to manage the File System Manager trace file.

▼ To Enable Tracing or To Adjust the Tracing Level

Use the following command to enable tracing or to adjust the tracing level:

```
# /opt/SUNWfsmgr/bin/fsmgr trace trace_level
```

For *trace_level*, specify one of the values shown in TABLE 3-2.

TABLE 3-2 Arguments for *trace_level*

trace_level	**Tracing Requested**
`off`	Disables tracing.
`1`	Enables tracing for very important messages only. This includes severe errors that occur within the application.
`2`	Enables tracing for moderately important messages. This includes level 1 messages as well as debugging statements within the application that are useful for developers.
`3`	Enables tracing for all messages. This includes level 1 and level 2 messages as well as entry and exit points of functions within the application on the stack.

You can enable and disable tracing dynamically during run time by using the `fsmgr`(1M) command.

File System Manager Portal Agent Tracing

The File System Manager Portal Agent acts as a information source for the Sun StorEdge Management Portal application.The agent uses the same tracing mechanisms as the File System Manager and both applications write to the same trace output file. Enabling tracing will enable tracing for both the File System Manager and the agent. Because of this, it can be hard to decipher the tracing output if both the agent and the File System Manager are running simultaneously. If you need to enable tracing, it is best to have either File System Manager or the agent running but not both.

If you enable tracing and do not see any tracing output for the agent, check the access permissions on the trace log file. The agent application runs as root, so you need to verify that root has access to write to the trace log file.

Remote Procedure Call (RPC) Daemon Information

The following procedure can help you obtain troubleshooting information for the RPC daemon, `fsmgmtd`(1M).

▼ To Determine Whether the RPC Daemon Is Running

Perform the following procedure to verify whether or not the RPC daemon is running and to obtain status information.

1. **Log in to the SAM-QFS server.**
2. **Become superuser.**
3. **Display status information for the File System Manager daemon (`fsmgmtd`).**

   Enter the following command to display the daemon:

   ```
   # /opt/SUNWsamfs/sbin/fsmadm status
   ```

   If the daemon is not running, it does not display its status. Enter the following command to start the daemon:

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

   The preceding command starts the daemon and enables it to automatically restart if it dies.

CHAPTER 4

Backing Up Data

This chapter provides the backup and dump processes and information you need in order to keep your data safe and prepare for any disaster.

This chapter includes the following subsections:

Guarding Against or Troubleshooting Data Loss

TABLE 4-1 shows the usual causes of data loss, with notes and suggestions about how to avoid or respond to each type of loss.

TABLE 4-1 Causes of Data Loss, With Notes and Suggestions

Causes	Notes	Suggestions
User Error	Sun StorEdge QFS file systems are protected from access by unauthorized users because of the UNIX superuser mechanism. You can also restrict administrative actions to an optional administrative group.	
System reconfiguration	File systems can be made unavailable by any of the following: • Dynamically-configured SAN components • Overwritten system configuration files • Failure of connectivity components	Rebuild the file system only after verifying that a configuration problem is not the cause of the apparent failure. See "Precautions Before Starting Data Restoration" on page 41 and "To Troubleshoot an Inaccessible File System" on page 41, and "Recovering From Catastrophic Failure" on page 117.
Hardware failure	Using disk storage systems managed by hardware RAID has the following advantages over systems managed using software RAID: • More reliability • Fewer resources are consumed on the host system • Better performance	Use hardware RAID disk storage systems wherever possible.
	Hardware-based inconsistencies in Sun StorEdge QFS file systems can be checked and fixed by unmounting the file system and running `samfsck`(1M) command.	Use `samfsck`(1M) to check and fix hardware-based file system consistency problems. See "To Troubleshoot an Inaccessible File System" on page 41 for an example. Also see "Recovering From Catastrophic Failure" on page 117.

Precautions Before Starting Data Restoration

Some apparent data losses are actually caused by cabling problems or configuration changes.

Caution – Do not reformat a disk, relabel a tape, or make other irreversible changes until you are convinced that the data on the disk or tape is completely unrecoverable.
Make sure to eliminate the fundamental causes for a failure before making irreversible changes. Back up anything you change before you change it, if possible.

Take the steps in the following procedure, "To Troubleshoot an Inaccessible File System," before commencing a data recovery process.

▼ To Troubleshoot an Inaccessible File System

1. **Check cables and terminators.**
2. **If you cannot read a tape or magneto-optical cartridge, try cleaning the heads in the drive, or try reading the cartridge in a different drive.**
3. **Check the current state of your hardware configuration against the documented hardware configuration.**

   Go to Step 4 only when you are certain that a configuration error is not to blame.
4. **Unmount the file system, and run** `samfsck`**(1M).**

   For example:

```
# umount file_system_name
# samfsck file_system_name
```

5. **If you find the file system is still inaccessible, use the procedures in the other chapters in this manual to restore the file system.**

Prerequisites for Data Recovery

For SAM-QFS file systems, the following are prerequisites for disaster recovery:

- Up-to-date archive copies

  The effectiveness of any of the SAM-QFS recovery methods relies primarily on frequent archiving being done.
- Up-to-date metadata dumps

  See "Metadata Used in Disaster Recovery" on page 42.
- Archiver logs

  If recent metadata is not available, archiver logs can help you recreate the filesystem directly from archive media.

  See "Using Archiver Logs" on page 60.

Note – Using archiver logs is much more time consuming that using metadata to retrieve data, so this approach should not be relied upon. It is not used unless there is no alternative.

Metadata Used in Disaster Recovery

Metadata consists of information about files, directories, access control lists, symbolic links, removable media, segmented files, and the indexes of segmented files. Metadata must be restored before lost data can be retrieved.

With the up-to-date metadata, the data can be restored as follows:

- File data can be restored even if the file has been removed from the file system.
- Individual files or entire file systems can be moved from one file system to another, or even from one server to another.

`.inodes` File Characteristics

In Sun StorEdge QFS file systems, the `.inodes` file contains all the metadata except for the directory namespace (which consists of the pathnames to the directories where the files are stored). The `.inodes` file is located in the root (`/`) directory of the file system. For a file system to be restored, the `.inodes` file is needed along with the additional metadata.

FIGURE 4-1 illustrates some characteristics of the `.inodes` file. The arrows with the dashed lines indicate that the `.inodes` file points to file contents on disk and to the directory namespace. The namespace also points back to the `.inodes` file. Also indicated is that, in SAM-QFS file systems where archiving is being done, the `.inodes` file also points to archived copies.

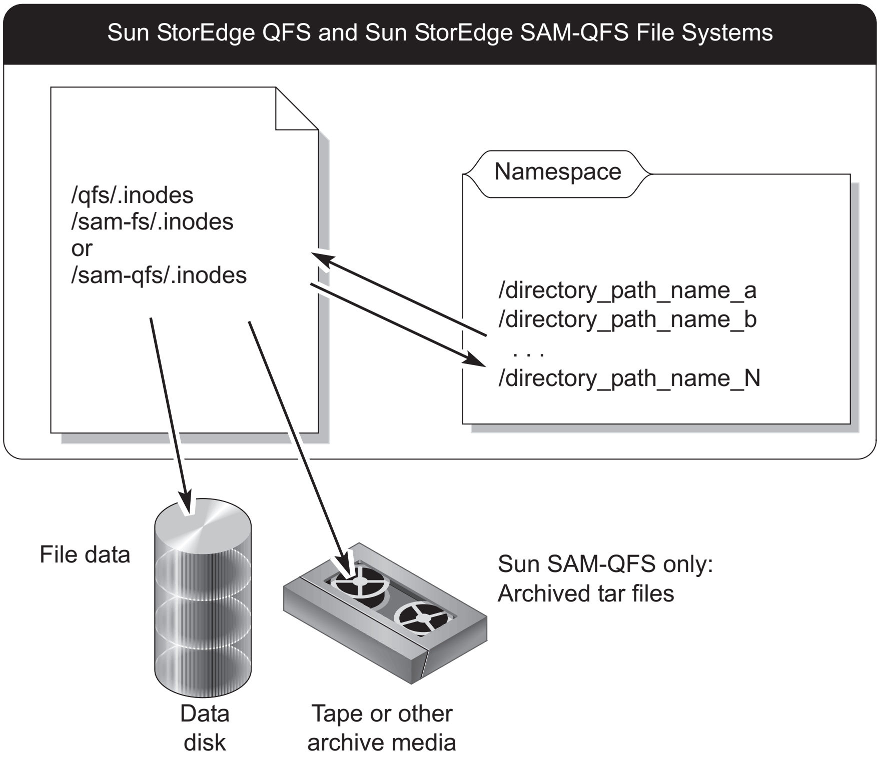

FIGURE 4-1 The .inodes File in Sun StorEdge QFS File Systems

Note – Sun StorEdge QFS has no archiving capability. See the *Sun StorEdge QFS Installation and Upgrade Guide* for how to back up Sun StorEdge QFS metadata.

The `.inodes` file is not archived. For more about protecting the `.inodes` file in these types of file systems, see “SAM-QFS Disaster Recovery Features” on page 45 and “Backing Up the Metadata in SAM-QFS File Systems” on page 47.

More About Directory Pathnames

As indicated in FIGURE 4-1, the namespace (in the form of directories) does not point to the archive media. The directory pathnames for each archived file *are* copied into the headers of the `tar`(1) files on the archive media that contain the files, *but* for reasons illustrated elsewhere (in TABLE 4-3), the directory pathnames in the `tar` file headers may get out of sync with the actual locations of the files on the disk.

One reason why the two pathnames can get out of sync is that the pathnames in the `tar` file header do not show the originating file system. TABLE 4-2 shows how the directory pathname shown in the left column would appear in the `tar` file header in the right column, without the component that shows the name of the originating file system `/samfs1`.

TABLE 4-2 Comparing a Full Pathname With a Pathname in a `tar` Header

Full Pathname	Pathname in `tar` Header on Archive Media
`/samfs1/dir1/filea`	`dir1/` `dir1/filea`

TABLE 4-3 summarizes an example scenario, shows the result, and suggests a precaution.

TABLE 4-3 Example of Potential Pitfalls

Scenario	Result	Precaution
File is saved to disk, archived, then later moved, either by use of the `mv`(1) command or by restoration from a `samfsdump`(1M) output file using `samfsrestore`(1M) into an alternate path or file system.	• Archive copy is still valid. • `.inodes` file still points to the archive media • Pathname in the `tar` file header no longer matches the namespace on disk. • Name of the file system is not available in the `tar` file header.	Keep the data from each file system on its own unique set of tapes or other archive media, and do not mix data from multiple file systems.

The potential for inconsistency does not interfere with recovery in most cases, because the directory pathnames in the `tar` headers are not used when data is being recovered from an archive. The directory pathnames on the `tar` headers on the archive media are only used in an unlikely disaster recovery scenario where no metadata is available and the file system must be reconstructed from scratch using the `tar` command.

SAM-QFS Disaster Recovery Features

The features of SAM-QFS file systems described in TABLE 4-4 streamline and speed up data restoration and minimize the risk of losing data in the case of unplanned system outage.

TABLE 4-4 Disaster Recovery Features of SAM-QFS File Systems

Feature	Comparison	Advantage
Identification records, serial writes, and error checking are dynamically used to check and manage file system consistency.	Eliminates the need to check file systems (by running the `fsck`(1M) command) before re-mounting the file systems or to rely on journal recovery mechanisms.	*Speed*. Because each file system is already checked and repaired when the server reboots after an outage, the server gets back into production more quickly.
Files are archived transparently and continuously. Archiving is configurable: after specified sleep intervals, via scheduled `cron`(1M) jobs, or on demand.	Nightly or weekly backups interfere with normal use of the system while the backups are being done and protection is not continuous.	*Data protection*. Because archiving is continuous, there are no gaps in data protection. Data backups no longer interfere with production.
Data can remain on disk or can be automatically released from the disk and then transparently staged back from archive media when needed.	Files no longer need to take up disk space. Files that are removed from the disk are instantly available without administrator intervention.	*Speed*. Disk space requirements may be lessened without inconvenience to users.
Files can be archived to as many as four separate media, each of which can be of a different type, and with Sun SAM-Remote, to remote locations.	Multiple copies can be easily made in multiple locations.	*Data Protection*. With the potential for multiple copies at multiple locations, the loss of one copy or even of an entire location does not mean a complete loss of data.

TABLE 4-4 Disaster Recovery Features of SAM-QFS File Systems *(Continued)*

Feature	Comparison	Advantage
Files are archived in standard `tar`(1) format files.	`tar` files can be restored onto any file system type.	*Flexibility*. SAM-QFS file systems do not need to be available.
Metadata can be restored separately from data. Restoration of the files' contents to disk is configurable: files can be staged only when they are accessed or in advance of anticipated need.	Restoring metadata allows users to access the system and their data without waiting until all data is restored to disk.	*Speed*. Access to the server is quicker than if all data needed to be restored before user access was allowed.

Guidelines for Performing Dumps

- Perform dumps with the file system mounted.
- Perform metadata dumps at a time when files are not being created or modified.

  At any given time, some files need to be archived because they are new, while others need to be rearchived because they are modified or because their archive media is being recycled. TABLE 4-5 defines the terms that apply to files archived onto archive media.

TABLE 4-5 Terms Related to Dumping Metadata

Term	When Used	Comments
stale	The archived copy does not match the online file.	A new copy must be created. Stale files can be detected using the `sls` command with the `-D` option. See the `sls`(1M) man page.
expired	No inode points to the archived copy.	A new archive copy was already created, and the file's inode correctly points to the new archive copy.

  Dumping metadata during a time when files are not being created or modified avoids the dumping of metadata for files that are stale and minimizes the creation of damaged files.

- If an error message identifies a file as damaged, run the `samfsdump`(1M) command again after the specified file is archived.

When any stale files exist while metadata and file data are being dumped, the `samfsdump` command generates a warning message. The following warning message is displayed for any files that do not have an up-to-date archive copy:

```
/pathname/filename: Warning! File data will not be recoverable (file
will be marked damaged).
```

Caution – If you see the above message and do not rerun the `samfsdump` command after the specified file is archived, the file will not be retrievable.

If `samfsrestore`(1M) is later used to attempt to restore the damaged file, the following message is displayed:

```
/pathname/filename: Warning! File data was previously not recoverable
(file is marked damaged).
```

Backing Up the Metadata in SAM-QFS File Systems

In SAM-QFS file systems, the `archiver`(1M) command can copy both file data and metadata—other than the `.inodes` file—to archive media. For example, if you create a SAM-QFS file system with a family-set name of `samfs1`, you can tell the `archiver` command to create an archive set also called `samfs1`. (See the `archiver.cmd`(4) man page for more information.) You can later retrieve damaged or destroyed file systems, files, and directories as long as the archive media onto which the archive copy was written has not been erased and as long as a recent metadata dump file is available.

The `samfsdump`(1M) command allows you to back up metadata separately from the file system data. The `samfsdump` command creates metadata dumps (including the `.inodes` file) either for a complete file system or of a portion of a file system. A `cron`(1M) job can be set up to automate the process.

If you dump metadata often enough using `samfsdump`, the metadata is always available to restore file data from the archives using `samfsrestore`(1M).

Note – Files written to the file system after metadata dumps begin might not be archived, and archive copies on cartridges might not be reflected in the metadata dump. Consequently, the files might not be known to the system if the dump is used to restore the file system. Files written to the file system or archived after the metadata dump are picked up during the next metadata dump.

In summary, using the `samfsdump` method to dump metadata has the following advantages:

- The `samfsdump` command saves the relative path for each file.
- The `samfsdump` command is run on mounted file systems.
- The metadata dump file generated by the `samfsdump` command contains all information required for restoring a SAM-QFS file system. The metadata dump file contains the `.inodes` file, directory information, and symbolic links.
- The `samfsdump` and `samfsrestore` method is flexible. This process enables you to restore an entire file system, a directory hierarchy, or a single file. With `samfsdump`(1M) and `samfsrestore`(1M), you can split an existing file system into multiple file systems or you can join multiple file systems into a single file system.
- The `samfsrestore` command defragments the `.inodes` file, the file system name space, and file data.

  During a file system restoration, files and directories are assigned new inode numbers based on directory location; only the required number of inodes are assigned. Inodes are assigned as the samfsrestore process restores the directory structure.

  File data is defragmented because files that were written in a combination of small disk allocation units (DAUs) and large DAUs are staged back to the disk using appropriately sized DAUs.
- When the `samfsrestore` process is complete, all directories and symbolic links are online and files are ready to be accessed.

Creating `samfsdump` Dump Files

If you have multiple SAM-QFS file systems, make sure that you routinely dump the metadata for every file system. Look in `/etc/vfstab` for all file systems of type `samfs`.

Make sure to save the dump for each file system in a separate file.

The following procedures describe how to find all the `samfs` type file systems and to dump metadata using `samfsdump`(1M):

- "To Find Sun StorEdge QFS File Systems" on page 50
- "To Create a Sun StorEdge SAM-FS Metadata Dump File Manually Using File System Manager" on page 50
- "To Create a Sun StorEdge SAM-FS Metadata Dump File Manually Using the Command Line" on page 51
- "To Create a Sun StorEdge SAM-FS Metadata Dump File Automatically From the File System Manager" on page 51
- "To Create a Sun StorEdge SAM-FS Metadata Dump File Automatically Using `cron`" on page 52

Note – The examples in these procedures use the names `/sam1` for a SAM-QFS file system mount point and `/dump_sam1` for the dump file system.

Using `samfsdump` With the `-u` Option

The `samfsdump`(1M) command `-u` option causes unarchived file data to be interspersed with the metadata. Note the following about the use of the `-u` option:

- A `samfsdump` command run with the `-u` option on a version 3.5 or 4.x SAM-QFS file system cannot be restored to an earlier version (3.3.*x*) file system of the same type because versions 3.5 and 4.x have new data structures. Dumps from a 4.x version of either file system type can be restored on a 3.5 version and vice versa.
- A `samfsdump` dump taken using the `-u` option can be very large. The `samfsdump` command does not have any tape management or estimations such as those associated with `ufsdump`(1M). You need to weigh the amount of dump storage space available against the risks of having unarchived data when using the `-u` option (as you do when setting up any data protection procedures). For more information, see also the `samfsdump` and `ufsdump` man pages.

To Find Sun StorEdge QFS File Systems

- **Look in the `vfstab`(4) file to find mount points for all samfs-type file systems.**

CODE EXAMPLE 4-1 shows three file systems of type `samfs` with the file system names `samfs1`, `samfs2`, and `samfs3`. The mount points are `/sam1`, `/sam2`, and `/sam3`.

CODE EXAMPLE 4-1 File Systems Defined in `/etc/vfstab`

```
# vi /etc/vfstab
samfs1 -       /sam1 samfs   -       no high=80,low=70,partial=8
samfs2 -       /sam2 samfs   -       no high=80,low=50
samfs3 -       /sam3 samfs   -       no high=80,low=50
```

To Create a Sun StorEdge SAM-FS Metadata Dump File Manually Using File System Manager

Taking a *metadata snapshot* through the File System Manager interface is the equivalent of using the `samfsdump` command from the command line. You can take a metadata snapshot from the File System Manager interface at any time.

To take a metadata snapshot:

1. **From the Servers page, click the server on which the file system that you want to administer is located.**

   The File Systems Summary page is displayed.

2. **Select the radio button next to the file system for which you want to schedule a metadata snapshot.**

3. **From the Operations menu, choose Take Metadata Snapshots.**

   The Take Metadata Snapshot pop-up window is displayed.

4. **In the Fully Qualified Snapshot File field, type the path and the name of the snapshot file that you want to create.**

Note – You must type the same path that is specified in the Snapshot File Path field on the Schedule Metadata Snapshot page for this file system. Otherwise, this snapshot file will not be displayed on the Restore File System page when you try to restore files for the file system.

---

5. **Click Submit.**

See the File System Manager online help file for complete information on creating metadata snapshots.

▼ To Create a Sun StorEdge SAM-FS Metadata Dump File Manually Using the Command Line

1. **Log in as root.**
2. **Go to the mount point for the `samfs` type file system mount point or to the directory that you are dumping.**

```
# cd /sam1
```

See "To Find Sun StorEdge QFS File Systems" on page 50 if needed.

3. **Enter the `samfsdump`(1M) command to create a metadata dump file.**

CODE EXAMPLE 4-2 shows a SAM-QFS file system metadata dump file being created on February 14, 2004 in a `dumps` subdirectory in dump file system `/dump_sam1/dumps`. The output of the `ls`(1) command line shows the date is assigned in the *yymmdd* format as the dump file's name, `040214`.

CODE EXAMPLE 4-2 Creating a Metadata Dump File

```
# samfsdump -f /dump_sam1/dumps/'date +\%y\%m\%d'
# ls /dump_sam1/dumps
040214
```

▼ To Create a Sun StorEdge SAM-FS Metadata Dump File Automatically From the File System Manager

Scheduling a metadata snapshot through the File System Manager interface is the equivalent of creating a `crontab`(1) entry that automates the Sun StorEdge SAM-FS software `samfsdump`(1M) process.

To schedule a metadata snapshot:

1. **From the Servers page, click the server on which the archiving file system that you want to administer is located.**

   The File Systems Summary page is displayed.

2. **Select the radio button next to the archiving file system for which you want to schedule a metadata snapshot.**
3. **From the Operations menu, choose Schedule Metadata Snapshots.**

   The Schedule Metadata Snapshots page is displayed.

4. **Specify values on the Schedule Metadata Snapshots page.**

   For complete instructions on using this page, see the File System Manager online help file.

5. **Click Save.**

▼ To Create a Sun StorEdge SAM-FS Metadata Dump File Automatically Using `cron`

1. **Log in as root.**

2. **Enter the `crontab`(1M) command with the `-e` option to make an entry to dump the metadata for each file system.**

   The `crontab` entry in CODE EXAMPLE 4-3 runs at 10 minutes past 2 a.m. every day and does the following:

   - In the dump file system's dumps directory (`/dump_sam1/dumps`), removes files older than three days.
   - Dumps the metadata from `/sam1`.
   - Assigns the date of the metadata dump as the file's name in *yymmdd* format.

CODE EXAMPLE 4-3 Crontab Entry

```
# crontab -e
10 2 * * * ( find /dump_sam1/dumps -type f -mtime +72 -print |
xargs -l1 rm -f; cd /sam1 ; /opt/SUNWsamfs/sbin/samfsdump -f
/dump_sam1/dumps/‘date +\%y\%m\%d ‘ )
:wq
```

Note – Make the `crontab` entry on a single line. Because the line in the previous screen example is too wide for the page's format, it breaks into multiple lines.

If the `crontab` entry in the previous screen example ran on March 20, 2005, the full pathname of the dump file would be: `/dump_sam1/dumps/050320`.

Disaster Recovery Commands and Tools

TABLE 4-6 summarizes the commands used most frequently in disaster recovery efforts.

TABLE 4-6 Disaster Recovery Commands and Tools

Command	Description	Used By
`qfsdump`(1M)	Dumps Sun StorEdge QFS file system metadata and data.	Sun StorEdge QFS
`qfsrestore`(1M)	Restores Sun StorEdge QFS file system metadata and data.	Sun StorEdge QFS
`samfsdump`(1M)	Dumps SAM-QFS file system metadata.	SAM-QFS
`samfsrestore`(1M)	Restores SAM-QFS file system metadata.	SAM-QFS
`star`(1M)	Restores file data from archives.	SAM-QFS

For more information about these commands, see their `man`(1) pages. Other scripts and helpful sample files are located `/opt/SUNWsamfs/examples` or are available from Sun Microsystems.

TABLE 4-7 describes some disaster recovery utilities in the `/opt/SUNWsamfs/examples` directory and explains their purpose. You must modify all of the listed shell scripts, except for `recover.sh`(1M), to suit your configuration before using them. See the comments in the files.

TABLE 4-7 Disaster Recovery Utilities

Utility	Description
`restore.sh`(1M)	Executable shell script that stages all files and directories that were online at the time a `samfsdump`(1M) was taken. This script requires that a log file generated by `samfsrestore`(1M) be used as input. Modify the script as instructed in the comments in the script. See also the `restore.sh`(1M) man page.
`recover.sh`(1M)	Executable shell script that recovers files from tape, using input from the archiver log file. If used with SAM-Remote clients or server, the recovery must be performed on the server to which the tape library is attached. For more information about this script, see the `recover.sh`(1M) man page and the comments in the script itself. Also see "Using Archiver Logs" on page 60.

TABLE 4-7 Disaster Recovery Utilities *(Continued)*

Utility	Description
`stageback.sh`	Executable shell script that stages files that have been archived on accessible areas of a partially damaged tape. Modify the script as instructed in the script's comments. For how the script is used, see "Damaged Tape Volume – No Other Copies Available" on page 103.
`tarback.sh`(1M)	Executable shell script that recovers files from tapes by reading each `tar`(1) file. Modify the script as instructed in the script's comments. For more information about this script, see the `tarback.sh` man page. See also "Unreadable Tape Label – No Other Copies Available" on page 105.

Caution – Improper use of the `restore.sh`, `recover.sh`, or `tarback.sh` scripts can damage user or system data. Please read the man pages for these scripts before attempting to use them. For additional help with using these scripts, contact Sun customer support.

The `samexplorer` Script

The `/opt/SUNWsamfs/sbin/samexplorer` script (called `info.sh` in software versions before 4U1) is not a backup utility, but it should be run whenever changes are made to the system's configuration.

The `samexplorer`(1M) script creates a file containing all the configuration information needed for reconstructing a SAM-QFS installation from scratch if you ever need to rebuild the system. You can use the `crontab`(1) command with the `-e` option to create a `cron`(1M) job to run the `samexplorer` script at desired intervals.

The `samexplorer` script writes the reconfiguration information to `/tmp/SAMreport`.

Make sure that the `SAMreport` file is moved from the `/tmp` directory after creation to a fixed disk that is separate from the configuration files and outside the SAM-QFS environment. For more information about managing the SAMreport file, see the `samexplorer`(1M) man page.

What to Back Up and How Often

TABLE 4-8 describes the files that should be backed up and how often the files should be backed up onto a location outside the file system environment.

Where “Regularly” is shown in the “Backup Frequency” column, each site’s system administrator should decide the appropriate intervals based on that site’s requirements. Except where specified, use whatever backup procedures you choose.

TABLE 4-8 Which Files to Back Up and How Often

Data Type	Backup Frequency	Comments
Site-modified versions of filesystem backup and restoration shell scripts.	After modification.	See the default scripts listed in “Disaster Recovery Commands and Tools” on page 53.
Site-created shell scripts and `cron`(1) jobs created for backup and restoration.	After creation and after any modification.	
`SAMreport` output from the `samexplorer`(1M) script.	At installation and after any configuration changes.	See the `samexplorer` script and `SAMreport` output file described in “The `samexplorer` Script” on page 54.
Sun StorEdge QFS metadata and data (see “Metadata Used in Disaster Recovery” on page 42 for definitions).	Regularly	Files altered after `qfsdump`(1M) is run cannot be recovered by `qfsrestore`(1M), so take dumps frequently. For more information, see “Metadata Used in Disaster Recovery” on page 42.
SAM-QFS metadata (see “Metadata Used in Disaster Recovery” on page 42 for definitions).	Regularly	Use the `samfsdump`(1M) command to back up metadata. Files altered after `samfsdump` is run cannot be recovered by `samfsrestore`(1M), so take dumps frequently or at least save the inodes information frequently. For more information, see “Backing Up the Metadata in SAM-QFS File Systems” on page 47.
SAM-QFS device catalogs.	Regularly	Back up all library catalog files, including the historian file. Library catalogs for each automated library, each pseudolibrary on Sun SAM-Remote clients, and for the historian (for cartridges that reside outside the automated libraries) are in `/var/opt/SUNWsamfs/catalog`.

TABLE 4-8 Which Files to Back Up and How Often *(Continued)*

Data Type	Backup Frequency	Comments
Archiver log files from a SAM-QFS file system where the archiver is being used.	Regularly	Specify a pathname and name for an archiver log file in the `archiver.cmd` file and back up the archiver log file. See the `archiver.cmd`(4) man page for how to specify an archiver log file for each file system. Also see “Using Archiver Logs” on page 60.
Configuration files and other similar files modified at your site. Note that these reside outside the SAM-QFS file system.	At installation and after any modification	The following files may be created at your site in the `/etc/opt/SUNWsamfs` directory: `archiver.cmd`(4) `defaults.conf`(4) `diskvols.conf`(4) `hosts.`*fsname* `hosts.`*fsname*`.local` `mcf`(4) `preview.cmd`(4) `recycler.cmd`(4) `releaser.cmd`(4) `rft.cmd`(4) `samfs.cmd`(4) `stager.cmd`(4)
Network-attached-library configuration files.	At installation and after any modification	If using network-attached libraries, make sure to back up the configuration files. The exact names of the files are listed in the `Equipment Identifier` field of the `/etc/opt/SUNWsamfs/mcf` file on each line that defines a network-attached robot. See the `mcf`(4) man page for more details.
Sun SAM-Remote configuration files.	At installation and after any modification	If using Sun SAM-Remote software, make sure to back up the configuration files. The exact names of the files are listed in the `Equipment Identifier` field of the `/etc/opt/SUNWsamfs/mcf` file on each line that defines a Sun SAM-Remote client or server. See the `mcf`(4) man page for more details.

TABLE 4-8 Which Files to Back Up and How Often *(Continued)*

Data Type	Backup Frequency	Comments

Installation files.	At installation and after any modification	The following files are created by the software installation process. If you have made local modifications, preserve (or back up) these files:
		`/etc/opt/SUNWsamfs/inquiry.conf`*
		`/opt/SUNWsamfs/sbin/ar_notify.sh`*
		`/opt/SUNWsamfs/sbin/dev_down.sh`*
		`/opt/SUNWsamfs/sbin/recycler.sh`*
		`/kernel/drv/samst.conf`*
		`/kernel/drv/samrd.conf`

Files modified at installation time.	At installation and after any modification	The following files are modified as part of the software installation process:
		`/etc/syslog.conf`
		`/etc/system`
		`/kernel/drv/sd.conf`*
		`/kernel/drv/ssd.conf`*
		`/kernel/drv/st.conf`*
		`/usr/kernel/drv/dst.conf`*
		Back the above files up so you can restore them if any of the files are lost or if the Solaris OE is reinstalled And if you modify the files, make sure to back them up again.

`SUNWqfs` and `SUNWsamfs` software packages and patches.	Once, shortly after downloading	The Sun StorEdge QFS and Sun StorEdge SAM software can be reinstalled easily from the release package and patches. Make sure you have a record of the revision level of the currently running software. If the software is on a CD-ROM, store the CD-ROM in a safe place. If you download the software from the Sun Download Center, back up the downloaded package(s) and patches. This saves time if you have to reinstall the software because you avoid having to download a fresh copy if you lose data.

TABLE 4-8 Which Files to Back Up and How Often *(Continued)*

Data Type	Backup Frequency	Comments
Solaris OS and patches; and unbundled patches.	At installation	The Solaris OE can be reinstalled easily from the CD-ROM, but make sure you have a record of all installed patches. This information is captured in the `SAMreport` file generated by the `samexplorer`(1M) script, which is described under "The `samexplorer` Script" on page 54. This information is also available from the Sun Explorer tool.

---

* Protect this file only if you modify it.

Additional Backup Considerations

The following is a list of questions to also consider when preparing your site's disaster recovery plan.

- What is the right number of `samfsdump`(1M) or `qfsdump`(1M) files to retain at your site?

  TABLE 4-9 compares the types of dumps that are done in the various file system types.

TABLE 4-9 Types of Dumps Performed on Sun StorEdge QFS Compared to SAM-QFS File Systems

File System Type	Dump Command Output	Notes
Sun StorEdge QFS	A `qfsdump`(1M) command generates a dump of both metadata and data.	See the *Sun StorEdge QFS Installation and Upgrade Guide* for how to back up Sun StorEdge QFS metadata.
SAM-QFS	The `samfsdump`(1M) command *without* the `-u` option generates a metadata dump file.	A metadata dump file is relatively small, so you should be able to store many more metadata dump files than data dump files. Restoration of the output of `samfsdump` without the `-u` option is quicker, because the data is not restored until accessed by a user.

TABLE 4-9 Types of Dumps Performed on Sun StorEdge QFS Compared to SAM-QFS File Systems *(Continued)*

File System Type	Dump Command Output	Notes
	The `samfsdump`(1M) command with the `-u` option dumps file data for files that do not have a current archive copy.	The dump files are substantially larger, and the command takes longer to complete. However, restoration of the output from `samfsdump` with `-u` restores the file system back to its state when the dump was taken.

Retain enough data and metadata to ensure that you can restore the file systems according to your site's needs. The appropriate number of dumps to save depends, in part, on how actively the system administrator monitors the dump output. If an administrator is monitoring the system daily to make sure the `samfsdump`(1M) or `qfsdump`(1M) dumps are succeeding (making sure enough tapes are available and investigating dump errors), then keeping a minimum number of dump files to cover vacations, long weekends, and other absences might be enough.

- If you are archiving data, are you actively recycling archive media? If so, make sure to schedule metadata copies to occur after recycling completes.

  If your site is using the `sam-recycler`(1M) command to reclaim space on archive media, it is critical that you make metadata copies *after* `sam-recycler` has completed its work. If a metadata dump is created before the `sam-recycler` exits, the information in the metadump about archive copies becomes out of date as soon as `sam-recyler` runs. Also, some archive copies may be made inaccessible because the `sam-recycler` command may cause archive media to be relabeled.

  Check root's `crontab`(1) entry to find out if and when the `sam-recycler` command is being run, and then, if necessary, schedule the creation of metadump files around the `sam-recycler` execution times. For more about recycling, see the *Sun StorEdge SAM-FS Storage and Archive Management Guide*.

- How much data should you store off site, and in what format?

  Off-site data storage is an essential part of a disaster recovery plan. In the event of a disaster, the only safe data repository might be an offsite vault. Beyond the recommended two copies of all files and metadata that you should be keeping in house as a safeguard against media failure, consider making a third copy on removable media and storing it offsite.

  Sun SAM-Remote offers you the additional alternative of making archive copies in remote locations on a LAN or WAN. Multiple Sun SAM-Remote servers can be configured as clients to one another in a reciprocal disaster recovery strategy.

- Is it sufficient to restore only the metadata to a predisaster state or do you need also to restore all files that were online when the disaster happened?

- The `samfsrestore`(1M) command can restore a SAM-QFS file or file system to the state reflected in the `samfsdump`(1M) file. After the `samfsrestore`(1M) command is run, the metadata is restored, but the file data remains offline.

  If you need to restore all files that were online, you need to run the `samfsrestore` command with the `-g` option.

  The log file generated by the `samfsrestore` command's `-g` option contains a list of all files that were on the disk when the `samfsdump`(1M) command was run. This log file can be used in conjunction with the `restore.sh` shell script to restore the files on disk to their predisaster state. The `restore.sh` script takes the log file as input and generates stage requests for files listed in the log. By default, the `restore.sh` script restores all files listed in the log file.

  If your site has thousands of files that need to be staged, consider splitting the log file into manageable chunks and running the `restore.sh` script against each of those chunks separately to ensure that the staging process does not overwhelm the system. You can also use this approach to ensure that the most critical files are restored first. For more information, see the comments in `/opt/SUNWsamfs/examples/restore.sh`.

Using Archiver Logs

Archiver logging should be enabled in the `archiver.cmd`(4) file. Because archiver logs list all the files that have been archived and their locations on cartridges, archiver logs can be used to recover lost files that were archived since the last set of metadata dumps and backup copies were created.

Be aware of the following considerations:

- Processes writing to the archiver log continue to do so until they complete.
- If a log file is not found, the SAM-QFS file system creates a new log file when a process initiates a new write to the log.
- If a log file exists, data is appended to the existing file.
- The archiver log files grow over time, so they must be managed.

Set up and manage the archive logs by performing the following procedures:

- "To Set Up Archiver Logging"
- "To Save Archiver Logs" on page 61

▼ To Set Up Archiver Logging

- **Enable archive logging in the `archiver.cmd` file (in the `/etc/opt/SUNWsamfs` directory).**

  See the `archiver.cmd`(4) man page. The archiver log files are typically written to `/var/adm/`*logfilename*. The directory where you direct the logs to be written should reside on a disk outside the SAM-QFS environment.

▼ To Save Archiver Logs

- **Ensure that archiver log files are cycled regularly by creating a `cron`(1M) job that moves the current archiver log files to another location.**

  The screen example below shows how to create a dated copy of an archiver log named `/var/adm/archlog` every day at 3:15 a.m. The dated copy is stored in `/var/archlogs`.

Note – If you have multiple archiver logs, create a `crontab` entry for each one.

```
# crontab -e
15 3 * * 0 (mv /var/adm/archlog /var/archlogs/‘date +%y%m%d‘; touch
/var/adm/archlog)
:wq
```

How and Where to Keep Copies of Disaster Recovery Files and Metadata

Consider writing scripts to create `tar`(1) files that contain copies of all the relevant disaster recovery files and metadata described in this chapter and to store the copies outside the file system. Depending on your site’s policies, put the files into one or more of the locations described in the following list:

- On another file system of any type.
- Directly on removable media files.

  For information on removable media files, see the `request`(1) man page.
- If running the `archiver`(1M) on a SAM-QFS file system, store the files on a separate SAM-QFS file system that is being archived on a separate set of

cartridges.

This approach ensures that the disaster recovery files and metadata are archived separately from file system to which they apply. You might also consider archiving multiple backup copies for additional redundancy.

Observe the following precautions:

- Keep a written (nonelectronic) listing of where the disaster recovery files are kept.

  You can obtain lists of all directories containing removable media files by using the `sls`(1M) command. These listings can be emailed. For more information about obtaining file information, see the `sls`(1M) man page.

- Keep a written record of your hardware configuration.
- Do not assign the cartridges used to hold the removable media files to the archiver.

CHAPTER 5

Restoring Files and Directories

This chapter describes how to restore individual files and directories. It covers the following topics:

TABLE 5-1 lists the tasks for restoring files and directories with cross references to where the procedures are located.

TABLE 5-1 Tasks for Restoring Files and Directories

Where Described	Notes
1. “Restoring Regular Files and Directories With `samfsdump`(1M) Output” on page 65[*]. 2. “Restoring Files and Directories Without `samfsdump`(1M) Output” on page 68[†]. 3. “Restoring Files Archived to Disk” on page 90. 4. “Retrieving Unarchived Files From File Systems” on page 99.	The first two procedures are for restoring files archived to tape or magneto-optical cartridges. These procedures are effective only if recent `samfsdump` files and recent archive copies of the files being restored are available.

[*] The same procedure is used for regular files, segmented files, volume overflow files, and directories.

[†] TABLE 5-2 points to the different procedures to use depending on whether the file is a regular file, a segmented file, a volume overflow file, or a file archived to disk.

Note – If the `mv`(1) command has been used to move an archived file to a different directory, the file is not rearchived. If you use the `star`(1M) command to recover a moved file, the `star`(1M) header on the archive media retains the original path name. When you use the `star`(1M) command to reload the file, the file is restored to its original location.

You can see the path by issuing the `star`(1M) command with its `tvbf` arguments. Then, you can extract the file to its original location by issuing the `star`(1M) command again. Finally, issue the `mv`(1) command to move the file to its new directory.

Restoring Regular Files and Directories With `samfsdump`(1M) Output

You can use one of the following procedures to restore files and directories by using the dump file created by `samfsdump`(1M).

Note – `samfsdump` and `samfsrestore` work on SAM-QFS file systems but not Sun StorEdge QFS stand-alone file systems. If needed, see the *Sun StorEdge QFS Configuration and Administration Guide*.

▼ To Restore Files Using File System Manager

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

   The File Systems Summary page is displayed.

2. **Select the radio button next to the file system for which you want to restore files.**

3. **From the Operations drop-down menu, choose Restore.**

   The Restore File System page is displayed.

4. **If the metadata snapshot file is displayed as a link in the Metadata Snapshot Summary table, skip to the next step. Otherwise, make the snapshot available by selecting the radio button next to the snapshot that is unavailable and click Make Available for Browsing.**

5. **Perform either of the following in the Metadata Snapshot Summary table:**

   - Click a metadata snapshot file to browse its contents.
   - Select the radio button next to a metadata snapshot file and click Browse.

   The Restore File System page is refreshed and the top-level items in the selected metadata snapshot are displayed in the Metadata Snapshot Entries table.

6. **Find the files that you want to restore.**

   For complete instructions on finding the files you want to restore, see the File System Manager online help file.

7. **In the Metadata Snapshot Entries table, select the radio button next to the file or directory that you want to restore.**

   The file or directory that you select is displayed in the File to Restore field.

8. **From the Online Status After Restoring drop-down menu, choose how you want the file restored.**

9. **Click Restore.**

Note – The File System Manager software can only restore files from snapshots created by the File System Manager software.

▼ To Restore Using a `samfsdump`(1M) File

The example in this procedure uses the `samfsrestore`(1M) command to restore a lost file from a dump file created by the `samfsdump` command. It restores a file (pathname: `/sam1/mary/mary1`) from a `samfsdump` metadata dump file called `/dump_sam1/041126`. The example creates a temporary restoration directory called `restore` in the `/sam1` file system.

1. **Use the `mkdir`(1) command to create a directory in which to restore the files within a SAM-QFS file system.**

```
# mkdir restore
```

2. **Use the `archive`(1) command with the `-r` option and `-n` option to prevent the archiver from archiving from this temporary directory location.**

```
# archive -r -n restore
```

3. **Use the `cd`(1) command to change to the temporary restoration directory.**

```
# cd restore
```

4. **Use the `samfsrestore`(1M) command with the `-t` and `-f` options to list the contents of the dump file.**

   After the `-f` option specify the dump file's path name as shown in CODE EXAMPLE 5-1.

CODE EXAMPLE 5-1 Listing the Content of a Dump File

```
# samfsrestore -t -f /dump_sam1/041126
samfsrestore -t -f /dump_sam1/041126
./lost+found
./neptune
```

CODE EXAMPLE 5-1 Listing the Content of a Dump File *(Continued)*

```
./mary
./fileA
./fileB
./fileC
./fileD
./fileE
./mary/mary1
./mary/mary2
./neptune/vmcore.0
./neptune/unix.0
./neptune/bounds
```

5. **Search the listing from the previous step to verify that the lost file is in the dump file.**

   If you find the file you are looking for, copy down the exact path name shown in the output to use in the following step.

   In the previous screen example, the lost file called `mary1` is shown as residing in the `./mary` directory.

6. **Use the `samfsrestore` command with the `-T` and `-f` options to restore the file's inode information to the current directory.**

   The *filename* must match exactly the pathname as it was listed in the previous output from Step 4. The following screen example shows using `samfsrestore` to retrieve the file `./mary/mary1` from the dump file `/dump_sam1/041126`.

```
# samfsrestore -T -f /dump_sam1/041126 ./mary/mary1
```

7. **Use the `sls(1)` command with the `-D` option to list detailed information about the file, and verify that the inode information for the correct file has been retrieved.**

   CODE EXAMPLE 5-2 shows inode information for file `./mary/mary1`.

CODE EXAMPLE 5-2 Verifying the inode Information

```
# sls -D ./mary/mary1
mary/mary1:
  mode: -rw-rw----  links:   1  owner: mary     group: sam
  length:        53  inode:          43
  offline;  archdone;
  copy 1: ---- Nov 17 12:35          8ae.1 xt 000000
  copy 2: ---- Nov 17 15:51       cd3.7f57 xt 000000
```

CODE EXAMPLE 5-2 Verifying the inode Information *(Continued)*

```
   access:       Nov 17 12:33  modification: Nov 17 12:33
   changed:      Nov 17 12:33  attributes:   Nov 17 15:49
   creation:     Nov 17 12:33  residence:    Nov 17 15:52
```

8. **Use the** `mv`**(1) command to move the file to the desired location.**

```
# cd mary
# mv mary1 /sam1/mary/
```

Restoring Files and Directories Without `samfsdump`(1M) Output

TABLE 5-2 lists the tasks for restoring various types of files when no `samfsdump`(1M) output is available.

TABLE 5-2 Tasks for Restoring Files When No `sampfdump`(1M) Output is Available

Type of File	Condition	Where Described
Regular file archived to removable media cartridges	An archiver log file exists with an entry for the file or you have output from the `sls` command with the `-D` option that lists the file.	• "To Restore a Regular File Using Information From an Archiver Log or `sls` Command Output" on page 72.
Regular file archived to removable media cartridges	No archiver log file exists	• "Restoring a Regular File Without Information From an Archiver Log" on page 75.

TABLE 5-2 Tasks for Restoring Files When No `sampfdump`(1M) Output is Available *(Continued)*

Type of File	Condition	Where Described
Regular file archived to disk	An archiver log file exists with an entry for the file or you have output from the `sls` command with the `-D` option that lists the file.	• "Restoring Files Archived to Disk" on page 90
Segmented file	An archiver log file exists with entries for the file.	• "Restoring a Segmented File Using Information From an Archiver Log" on page 82.
Volume overflow file	An archiver log file exists with entries for the file.	• "Restoring a Volume Overflow File Using Information From an Archiver Log" on page 88.

When you have an archiver log with an entry or entries for a missing file, see the following sections for how to interpret the information in the archiver log file and how to determine which of the above procedures to use:

- "Information Needed to Restore a File" on page 70
- "Determining Whether a File Is a Regular File, a Segmented File, or a Volume Overflow File" on page 70

Note that if you restore files without `samfsdump`(1M) output, you recreate the `.inodes` file. In addition, you lose the content of the original `.inodes` file. If you had used `chmod`(1), `chown`(1), or any other commands to modify file attributes, those attributes are lost. The files are restored with their default attributes. This applies to all types of files (regular files, segmented files, and so on).

Information Needed to Restore a File

TABLE 5-3 shows the information needed for restoring a regular file.

TABLE 5-3 Information Needed for Restoring a Regular File

Definition	Field in Archiver Log Output	Field in Archive Copy Line in `sls -D` Output
Media type	4	5
Volume Serial Name (VSN)	5	6
Position*	7	4

* The position is the value on the left of the field with the format: *position.offset*.

If you can get the needed information about a regular file either from its archiver log entry or from output about the file from the `sls`(1) command with the `-D` option, you can restore the file with the `request`(1M) and `star`(1M) commands. As shown in the examples that follow, the `request` command is first used to create a file whose contents represent the contents of one or more pieces of removable media (which is sometimes referred to as a *request file*). The `star` command is then used to extract the file, as shown in "To Restore a Regular File Using Information From an Archiver Log or `sls` Command Output" on page 72.

Determining Whether a File Is a Regular File, a Segmented File, or a Volume Overflow File

This section shows how to determine from a missing file's archiver log file entries whether the file is a regular file, a segmented file, or a volume overflow file. You need this information to decide which of the restoration procedures to follow from "Restoring Files and Directories Without `samfsdump`(1M) Output" on page 68.

Regular File

Each *regular* file has a single entry in an archiver log. In field 12 of the archiver log entry a regular file is identified with an `f`. The following example shows a typical entry for a regular file in an archiver log:

```
A 96/01/05 10:55:56 mo v1 set_1.1 d2e.1 samfs2 770.11 2673 test/file3 f 0 0
```

Segmented File

A *segmented* file is a file that has the segment attribute set and a *segment_size* specified using the `segment`(1) command. When a file has the segment attribute set, it is archived and staged in segment-sized chunks. The length of the segment (segment_size) is shown in field 10 of the archiver log file in kilobytes.

For each segmented file, an archiver log has multiple entries. CODE EXAMPLE 5-3 shows three entries for segmented file `seg/aaa`. Field 12 has a S indicating that the file type is *file segment*.

CODE EXAMPLE 5-3 Archiver Log Entry for a Segmented File

```
A 2000/06/15 17:07:28 ib E00000 all.1 1276a.1 samfs4 14.5 10485760
seg/aaa/1 S 0 51
A 2000/06/15 17:07:29 ib E00000 all.1 1276a.5002 samfs4 15.5
10485760 seg/aaa/2 S 0 51
A 2000/06/15 17:07:29 ib E00000 all.1 1276a.a003 samfs4 16.5 184
seg/aaa/3 S 0 51
```

Volume Overflow File

A volume overflow file is one that is written on multiple volumes. For a volume overflow file, an archiver log has multiple entries, one for each section of the file. CODE EXAMPLE 5-4 shows two entries for the two sections of file `big2d`.

CODE EXAMPLE 5-4 Archiver Log Entry for a Volume Overflow File

```
A 2001/10/31 09:47:29 lt CFX600 arset1.1 3668e.1 samfs9 71950.15
2011823616 testdir1/big2d f 0 43
A 2001/10/31 09:47:29 lt CFX603 arset1.1 3844a.0 samfs9 71950.15
1209402048 testdir1/big2d f 1 41
```

The `big2d` file is identified as a volume overflow file because it has two entries, the `f` in field 12 indicates that the entry is for a regular file, and the `0` and the `1` in field 13 are section numbers. Field 5 shows that the file starts on VSN `CFX600` and overflows to VSN `CFX603`.

Summary of Differences

TABLE 5-4 summarizes the defining characteristics of regular, segmented, and volume overflow files.

TABLE 5-4 Defining Characteristics of Regular, Segmented, and Volume Overflow Files

File Type	Definition
A file is a regular file if . . .	It has only a single entry The file type in field 12 is `f`.
A file is a segmented file if . . .	It has multiple entries. The VSN in field 5 is the same for each entry. The file type in field 12 is S. The section numbers in field 13 are the same for each entry.
A file is a volume overflow file if . . .	It has multiple entries. The VSN in field 5 is different for each entry. The file type in field 12 is `f`. The section numbers in field 13 are different for each entry.

▼ To Restore a Regular File Using Information From an Archiver Log or `sls` Command Output

---

Note – For the procedure to work, the SAM-QFS file system must be mounted.

---

1. **Log in as, or switch user to, root.**
2. **Find the media type, the file's position, and the VSN.**

a. **If you have an archiver log, use `cat`(1M) or another command to search the archiver log file for an entry for the missing file.**

   CODE EXAMPLE 5-5 shows the sample entry for a file that is archived on a tape followed by a sample entry for a file archived on an optical disk.

CODE EXAMPLE 5-5 Using the `cat`(1) Command to Search the Archiver Log

```
# cat
...
A 96/06/04 10:55:56 lt DLT001 arset0.1 286.1324f samfs1 770.11
130543 tape_test/file4 0 0 0
A 96/01/05 10:55:56 mo v1 set_1.1 d2e.1 samfs2 770.11 2673
test/file3 0 0 0
```

   If needed, see TABLE 5-3 for definitions of the fields in the archiver log file.

b. **If you have output from the `sls` command with the `-D` option about the missing file, search that output.**

   CODE EXAMPLE 5-6 shows output from the `sls`(1) command with the `-D` option for file `tape_test/file4`.

CODE EXAMPLE 5-6 `sls`(1) Output using the `-D` Option

```
# sls -D /sam1/tape_test/file4
/sam1/tape_test/file4:
 mode: -rw-rw----  links:   1  owner: root      group: other
 length:    130543
 offline;
 copy 1:   Jun  4 10:55       286.1324f lt DLT001
 access:   May 24 16:55  modification: May 24 16:38
 changed:  May 24 16:38  attributes:   Jun  4 10:55
 creation: May 24 16:38  residence:    Jun  4 10:55
```

c. **Record the media type, the file's position, and the VSN to use as input to the `request`(1M) command in the next step.**

3. **Use the `request`(1M) command with the `-p` option using the position from the archiver log to position to the beginning of the `tar`(1) header for the file.**

   Use hexadecimal notation, prefacing the position number after the `-p` option with `0x`.

   The following example creates a request file with the contents of the archive containing the example file that is on tape:

```
# request -p 0x286 -m lt -v DLT001 /sam1/xxxx
```

The following example creates a request file with the contents of the example file that is on optical disk:

```
# request -p 0xd2e -m mo -v v1 /sam2/xxxx
```

Note – VSNs specified using `request`(1M) command must reside on a local automated library.

---

4. **Use the `star`(1M) command to extract the file as shown in** CODE EXAMPLE 5-7**.**

Note – The `star`(1M) command restores all the files from the archive file that you are pointing to with the request file.

CODE EXAMPLE 5-7 Using `star`(1M) to Restore Files

```
# cd /sam1
# star -xv -b 32 -f /sam1/xxxx

...
tape_test/file4
...
tar: directory checksum error

# cd /sam2
# star -xv -b 32 -f /sam2/xxxx
...
test/file3
...
tar: directory checksum error
#
```

Note – You can ignore the directory checksum error.

If you labeled the tape with a block size other than the default (16 kilobytes), you would use the block size in bytes divided by 512 (in place of the value 32) for the `star` command's `-b` option. You can see the tape block size by mounting the tape and observing either the `samu`(1M) utility's `t` display, the `samu` utility's `v` display (type CTRL-i for detail lines), or the output of the `dump_cat`(1M) command.

5. **Use the `sls`(1) command to verify that you have extracted the lost file.**

CODE EXAMPLE 5-8 shows the `sls -D` output for the file on the optical disk.

CODE EXAMPLE 5-8 Using `sls`(1) to Verify the Extracted Files

```
# sls -D /sam2/test/file3
/sam2/test/file3:
mode: -rw-rw----   links:    1   owner: root       group: other
length:        2673  admin id: 7  inode:       161.2
copy 1:---- May   1 15:41        286.1324f  mo  v1
access:    May   1 16:50   modification: May   1 15:41
changed:   May   1 15:40   attributes:    May   1 15:44
creation:  May   1 15:40   residence:     May   1 16:50
```

Restoring a Regular File Without Information From an Archiver Log

If you do not have an archive log available with an entry for the file, you can use the procedure “To Restore a Regular File Without Information From an Archiver Log” on page 76.

Note – If the only resources available consist of a cartridge containing archive copies and a Solaris system without Sun StorEdge SAM-FS software installed, you can still restore the file by starting this procedure with Step 3.

You can perform the procedure “To Restore a Regular File Without Information From an Archiver Log” on page 76 using either an automated library or a manually mounted, standalone drive, under the following conditions:

- If you are using an automated library, the automated library daemon must be active on the system.
- If you are using a manually mounted, standalone drive, make sure that `/kernel/drv/st.conf` is correctly configured for the tape drive that you are using. For more information about performing this task, see how to add tape support to the `st.conf` file in the *Sun StorEdge SAM-FS Installation and Upgrade Guide*.

To determine which cartridge contains the missing file, you need to examine only those volumes that are assigned to the archive set for the file in question. You can use the `-t` option to `tar` or `star` repeatedly on each volume as described in the procedure “To Restore a Regular File Without Information From an Archiver Log” on page 76 to find out which volume contains the archive copy. When you have found the archive copy of the file, you then use the `-x` option to `tar` or `star` to extract the file.

▼ To Restore a Regular File Without Information From an Archiver Log

1. **(Optional) Prevent the Sun StorEdge SAM-FS software from using the tape drive**.

Note – If you are using a manually mounted, standalone drive, skip this step.

You can use either the `samu`(1M) command with the `:unavail` *eq* option, the `samcmd`(1M) command with the `unavail` *eq* option, the `devicetool`(1M) or the `libmgr`(1M) command. For the `samu` and `samcmd` commands, specify the equipment ordinal of the drive as *eq*. The Equipment Ordinal for each device is specified in the `mcf`(4) file.

The following screen example shows the use of the `samcmd` command with the `unavail` subcommand when the equipment ordinal is 51.

```
# samcmd unavail 51
```

2. **(Optional) Use the `samload`(1M) command to load the desired volume into the drive.**

Note – If you are using a manually mounted, standalone drive, skip this step.

For the command line options to use, see the `samload`(1) man page. The following screen example shows the use of the `samload` command to load the cartridge that is in slot 3 of library 50 into the drive with equipment ordinal 51

```
# samload 50:03 51
```

3. **Use the `mt`(1M) command to rewind the tape.**

The following example shows how to do this using the `mt`(1M) command. If your tape drive is not `/dev/rmt/2`, substitute the correct name in the following examples.

```
# mt -f /dev/rmt/2cbn rewind
```

Note – Because the device name used in these examples ends with the `n` (no rewind) option, each of the commands in the following steps examines the next file on the tape.

---

4. **Use `od`(1M) or another command to examine the ANSI label on the cartridge, and find the line that starts with `0000240`.**

   The first file on the cartridge is the ANSI label. In the following example, the information you are looking for appears on the line that starts with `0000240`.

```
# od -c /dev/rmt/2cbn
0000000   V   O   L   1   X   X   X
0000020                                   S   A   M   -   F   S       1
0000040   .   0
0000060
0000100                                                               4
0000120   H   D   R   1
0000140                                               0   0   0   1   0
0000160   0   0   1   0   0   0   1   0   0       2   4   9   0   9
0000200                                                   S   A   M   -
0000220   F   S       1   .   0
0000240   H   D   R   2       1   6   3   8   4                   1
0000260                                                   2   0   g 031
0000300
*
0000360
```

5. **Note the five characters that appear after `H D R 2` on the line that starts `0000240`.**

   The five characters that appear after `H D R 2` on the line that starts with `0000240` are the five bottom digits of the block size, in decimal. In the previous screen example, the characters are `1 6 3 8 4`.

6. **Use the five bottom digits of the block size to determine the block size used on the media.**

   Locate the bottom five digits of the block size in the left column of TABLE 5-5. For the `dd`(1M) command, the block size is found in the second column. For both the `star`(1M) and `tar`(1) commands, the block size is specified in units of 512-byte blocks, which are shown in column 3.

TABLE 5-5 Block Sizes Corresponding to the Bottom Five Digits of Block Size in the ANSI Label

Bottom Five Digits of Block Size	**Block Size for `dd`(1)**	**512-byte Blocks for `tar`(1) and `star`(1M)**
16384	16 kilobytes	32 blocks
32768	32 kilobytes	64 blocks
65536	64 kilobytes	128 blocks
31072	128 kilobytes	256 blocks
62144	256 kilobytes	512 blocks
24288	512 kilobytes	1024 blocks
48576	1024 kilobytes	2048 blocks
97152	2048 kilobytes	4096 blocks

Note – In the following screen examples, all files are archived twice, so each file is inspected twice.

---

7. **If the `star`(1M) command is available, enter it with the number of 512-byte blocks obtained in the previous two steps to find the file in the archive.**

   You can download the `star` command from a Sun StorEdge SAM-FS system onto any Solaris system. If you do not have access to the `star` command, you can use the `dd`(1M) command with the `tar`(1) command, as shown in Step 8.

Note – `star` files have an extended maximum file size of 1 Tbytes-1. `tar` and `star` files have compatible formats only at file sizes less than or equal to (≤) 8Gbytes-1. At larger than (≥) 8Gbytes, the formats of `star` and `tar` files are not compatible. Therefore, you must use the `star` command to read archives larger than 8 Gbytes-1.

CODE EXAMPLE 5-9 shows the `star` command being used to examine the first `tar` file. The block size for both the `star`(1M) and `tar`(1) commands is specified in units of 512-byte blocks. (The number 32 used after `-b` in the example is the number of 512-byte blocks that corresponds to the number 16384 in the ANSI label in Step 4, from the table in Step 6.

CODE EXAMPLE 5-9 `star`(1M) Command for Examining First `tar`(1) File

```
# star -tv -b 32 -f /dev/rmt/2cbn
-rw-rw----  0/1   102564 Sep  6 13:02 1996 test
6+1 records in
11+1 records out
```

CODE EXAMPLE 5-10 shows the same command examining the next `tar`(1) file.

CODE EXAMPLE 5-10 `star`(1M) Command for Examining Second `tar`(1) File

```
# star -tv -b 32 -f /dev/rmt/2cbn
-rw-rw----  0/1   102564 Sep  6 13:02 1996 test
6+1 records in
11+1 records out
```

CODE EXAMPLE 5-11 shows two copies of another file being examined.

CODE EXAMPLE 5-11 `star`(1M) Command for Examining Additional `tar`(1) Files

```
# star -tv -b 32 -f /dev/rmt/2cbn
-rw-rw----  0/1   102564 Sep  6 13:02 1996 test2
6+1 records in
11+1 records out
# star -tv -b 32 -f /dev/rmt/2cbn
-rw-rw----  0/1   102564 Sep  6 13:02 1996 test2
6+1 records in
11+1 records out
```

CODE EXAMPLE 5-12 shows the end of the tape has been reached.

CODE EXAMPLE 5-12 star(1M) and mt(1M) Output Showing the End of the Tape

```
# star -tv -b 32 -f /dev/rmt/2cbn
0+0 records in
0+0 records out
tar: blocksize = 0
# mt -f /dev/rmt/2cbn status
Other tape drive:
 sense key(0x13)= EOT   residual= 0   retries= 0
 file no= 5   block no= 0
```

8. **If the `star`(1M) command is not available, use the `dd`(1M) and `tar`(1) commands to examine the archives.**

   CODE EXAMPLE 5-13 shows the `dd` command being used to examine the first `tar` file. The value 16k used for the input block size (`ibs=`) is the number in the third column of the table in Step 6 that corresponds to the number 16384 in the ANSI label in Step 4.

   CODE EXAMPLE 5-13 `dd`(1M) Command for Examining the First `tar`(1) File

```
# dd if=/dev/rmt/2cbn ibs=16k obs=10k conv=sync | tar tvf -
-rw-rw----  0/1   102564 Sep  6 13:02 1996 test
6+1 records in
11+1 records out
```

   CODE EXAMPLE 5-14 shows the same command examining the next `tar`(1) file.

   CODE EXAMPLE 5-14 `dd`(1M) Command for Examining the Next `tar`(1) File

```
# dd if=/dev/rmt/2cbn ibs=16k obs=10k conv=sync | tar tvf -
-rw-rw----  0/1   102564 Sep  6 13:02 1996 test
6+1 records in
11+1 records out
```

   CODE EXAMPLE 5-15 shows the examination of two copies of another file.

   CODE EXAMPLE 5-15 `dd`(1M) Command for Examining Additional `tar`(1) Files

```
# dd if=/dev/rmt/2cbn ibs=16k obs=10k conv=sync | tar tvf -
-rw-rw----  0/1   102564 Sep  6 13:02 1996 test2
6+1 records in
11+1 records out
# dd if=/dev/rmt/2cbn ibs=16k obs=10k conv=sync | tar tvf -
-rw-rw----  0/1   102564 Sep  6 13:02 1996 test2
6+1 records in
11+1 records out
```

CODE EXAMPLE 5-16 shows that the end of the tape has been reached.

CODE EXAMPLE 5-16 `dd`(1M) and `mt`(1M) Output Showing the End of the Tape

```
# dd if=/dev/rmt/2cbn ibs=16k obs=10k conv=sync | tar tvf -
0+0 records in
0+0 records out
tar: blocksize = 0
# mt -f /dev/rmt/2cbn status
Other tape drive:
 sense key(0x13)= EOT   residual= 0   retries= 0
 file no= 5   block no= 0
```

Note – You might receive errors during this process. The following error indicates that the block size you selected does not match that of the tape:

```
read: not enough space
```

Correct the block size and try again.

---

9. **When you find the missing file in an archive, use the `-x` option with either the `star` command alone or the `dd` command with the `tar` command to extract the files from that archive.**

CODE EXAMPLE 5-17 shows these commands.

Note – You can ignore the `dd: read error` in the first line of output.

CODE EXAMPLE 5-17 Using the `star`(1M) command or the `dd`(1M) and `tar`(1) Commands

```
# dd if=/dev/samst/c0t1u0 bs=1k iseek=3374 of=/tmp/junk count=10
dd: read error: I/O error
8+0 records in
8+0 records out
# tar xvf /tmp/junk

# star -xv -f /tmp/junk
tar: blocksize = 1
-rw-rw---- 0/1 2673 May 1 15:41 1996 dir3/dir2/file0
-rw-rw---- 0/1 946 May  1 15:41 1996 dir3/dir1/file1
-rw-rw---- 0/1 468 May  1 15:41 1996 dir1/dir3/file0
```

Restoring a Segmented File Using Information From an Archiver Log

When a segmented file is archived or staged, it is archived and staged in chunks. For each segmented file, an archiver log has multiple entries.

If an archiver log file exists, you can search the archiver log for the multiple entries for the missing segmented file. (See "To Set Up Archiver Logging" on page 61, if needed.)

If you can find entries for a missing segmented file in an archiver log, you can use the file's position, segment size, VSN, and media type, to restore the file using the `request`(1M) and `star`(1M) commands. The procedure is described in "To Restore a Segmented File Using Information From Archiver Log Entries" on page 83.

If needed, see TABLE 5-3 for definitions of the fields in the archiver log file.

The segmented file named `aaa` is used in the examples in this section and in the procedure. CODE EXAMPLE 5-18 shows three entries for segmented file `aaa` in the archiver log file.

CODE EXAMPLE 5-18 Archiver Log File Showing Segmented File `aaa`

```
A 2000/06/15 17:07:28 ib E00000 all.1 1276a.1 samfs4 14.5 10485760
seg/aaa/1 S 0 51
A 2000/06/15 17:07:29 ib E00000 all.1 1276a.5002 samfs4 15.5
10485760 seg/aaa/2 S 0 51
A 2000/06/15 17:07:29 ib E00000 all.1 1276a.a003 samfs4 16.5 184
seg/aaa/3 S 0 51
```

Note – CODE EXAMPLE 5-18 assumes that all segments are in the same tape `tar`(1) file and that no segment is overflowed. If there are segments on more than one `tar`(1) file, use a separate `request`(1M) command for each `tar`(1) file position. If any segments are volume overflow files, use the procedure described in "Restoring a Volume Overflow File Using Information From an Archiver Log" on page 88 for those segments.

TABLE 5-6 provides a place to record the information used when restoring a segmented file.

TABLE 5-6 Archiver Log Entry Information Needed for Restoring a Segmented File

Field	Definition	Notes
4	Media type	
5	VSN	
7	Position	
12	File type	The `S` in field 12 indicates that the entry is for a segment of a segmented file.
11	File name	In the file name field of the three example entries, the three segments of the file `aaa` are identified as `seg/aaa/1`, `seg/aaa/2`, and `seg/aaa/3`.
10	Length	For entries for file segments, the segment size (length) is shown. You specify the segment size of the first segment on the `segment(1)` command line to recover a segmented file.

▼ To Restore a Segmented File Using Information From Archiver Log Entries

---

Note – Free space must be available in the file system equal to two times the size of the file to be recovered.

---

1. **Find the archiver log entries for the segmented file by the file system name (from field 8) and file name (from field 11).**

   CODE EXAMPLE 5-19 shows entries for segmented file `file2` in the `archiver.log` file.

CODE EXAMPLE 5-19 Example Archiver Log File

```
A 2002/11/19 14:01:47 ib E00000 all.1 1276a.1 samfs4 14.5 10485760 seg/aaa/1 S
0 51
A 2002/11/19 14:04:11 ib E00000 all.1 1276a.5002 samfs4 15.5 10485760 seg/aaa/2
S 0 51
A 2002/11/19 14:06:24 ib E00000 all.1 1933a.1 samfs4 16.5 184 seg/aaa/3 S 0 51
```

CODE EXAMPLE 5-19 shows the archiver log file for file segments in file system `qfs1`. Each segment has its own entry and file name: `seg/aaa/1`, `seg/aaa/2`, `seg/aaa/3`, and so on.

If needed, see TABLE 5-3 for definitions of the fields in the archiver log file.

2. **Note the contents of several fields in the archiver log.**

   Information from the archiver log needs to be used as input to the `request`(1M) command in Step 3 and to the `segment(1)` command in Step 9. The information needed is contained in the following fields:

   - Field 4. Media type on which the file is stored. For the supported media types, see the `mcf`(4) man page.
   - Field 5. VSN.
   - Field 7. File position. From the position indicator portion to the left of the period (`.`) in the field.
   - Field 10. Segment size. This is the length field.

   The first line in the previous screen example provides the following information:

   - The media type is `ib`.
   - The VSN is `E00000`.
   - The file's position is `1276a`.
   - The segment size is `10485760`.

3. **Enter the `request`(1M) command to create a removable media file that points to the segments.**

   Supply the following information as arguments to the `request`(1M) command:

   - *media type* after the `-m` option.
   - *position* number after the `-p` option in hexadecimal notation. Preface the position number with `0x`.
   - *VSN* after the -v option.
   - *filename* for a removable media file.

   The following command uses the values from the example lines in Step 1:

```
# request -m ib -p 0x1276a -v E00000 /sam3/rmfile
```

   The preceding command retrieves the first two segments.

Note – VSNs specified using `request`(1M) command must reside on a local automated library.

---

4. **Enter the `star`(1M) command, as shown in CODE EXAMPLE 5-20.**

   Use the name of the file created in the previous step to read the segments from tape onto the disk.

CODE EXAMPLE 5-20 Read the Segments from Tape to Disk

```
# star xvbf 512 /sam3/rmfile
seg/aaa/1
seg/aaa/2
```

5. **Repeat Step 2, Step 3, and Step 4 for each segment or group of segments that is located at a unique position (even if they are on the same VSN).**

   There might be more than one segment at a position. You might need to issue this step, Step 3, and Step 4 as a sequence more than once.

   If there are segments on different VSNs, make sure you specify the correct media type and VSN for each.

   The following commands complete the command sequence that is needed to retrieve the three file segments described in CODE EXAMPLE 5-19:

CODE EXAMPLE 5-21 Entering Additional Commands

```
# request -m ib -p 0x1933a -v E00000 /sam3/rmfile
# star xvbf 512 /sam3/rmfile
```

The commands in CODE EXAMPLE 5-21 are needed because the third segment is at a different position from the first two segments. If there are file segments at more than one location, you must enter `request`(1M) and `star`(M) commands to retrieve all the segments. You can dispense with other files that might be retrieved in the same tarball.

6. **Use the `cd`(1) command to change into the directory where the segmented files reside.**

   CODE EXAMPLE 5-22 shows segmented files `1`, `2`, and `3` in the `seg/aaa` directory.

CODE EXAMPLE 5-22 Change to the Directory Where the Segmented Files Reside

```
# cd seg
# pwd
/sam3/seg
# ls -l
total 8
drwxrwx---   2 root     other       4096 Jun 15 17:10 aaa/
# ls -l aaa
total 40968
-rw-rw----   1 root     other    10485760 Jun 15 17:06 1
```

CODE EXAMPLE 5-22 Change to the Directory Where the Segmented Files Reside

```
-rw-rw----   1 root     other    10485760 Jun 15 17:06 2
-rw-rw----   1 root     other         184 Jun 15 17:07 3
# pwd
/sam3/seg
# cd aaa
# pwd
/sam3/seg/aaa
```

7. **Use the `ls`(1) and `sort`(1) commands to list and sort the numbered files in numerical order, and use the `cat`(1M) command to join the files.**

   The temporary file created in this step is not segmented.

```
# ls | sort -n | xargs cat > ../bbb
```

8. **Use the `cd`(1) command to change to the directory above where the numbered files reside, and then use the `rm`(1) command to remove the numbered files.**

   CODE EXAMPLE 5-23 shows this.

CODE EXAMPLE 5-23 Change to the Directory Above the Numbered Files

```
# cd ..
# pwd
/sam3/seg
# ls -l
total 41000
drwxrwx---   2 root     other        4096 Jun 15 17:10 aaa/
-rw-rw----   1 root     other    20971704 Jun 15 17:11 bbb
# ls -l aaa
total 40968
-rw-rw----   1 root     other    10485760 Jun 15 17:06 1
-rw-rw----   1 root     other    10485760 Jun 15 17:06 2
-rw-rw----   1 root     other         184 Jun 15 17:07 3
# rm -rf aaa
```

9. **Enter the `touch`(1M) command to create an empty file.**

```
# touch aaa
```

10. **Use the `segment(1)` command to set the segment attribute on the file created in Step 9.**

    Enter the `segment` command with the `-l` option following by the segment length in megabytes followed by `m` followed by the filename of the empty file created in the previous step.

    Convert the segment length (from field 10 of the archiver log file entry) to megabytes by dividing 1048576. For example, the segment length in the archiver log entry example in Step 2 is 10485760. Dividing the segment length by 1048576 gives 10 megabytes, which is entered as `-l 10m` in the following screen example.

```
# segment -l 10m aaa
```

11. **Copy the temporary file created in Step 7 into the empty file created in Step 9, and then remove the temporary file.**

    CODE EXAMPLE 5-24 shows this procedure.

CODE EXAMPLE 5-24 Copying the Temporary File and then Removing It

```
# cp bbb aaa
# rm bbb
```

12. **Enter the `sls`(1) command with the `-2K` option to list the segments of the segmented file in two lines of output.**

    CODE EXAMPLE 5-25 shows this procedure.

CODE EXAMPLE 5-25 Using the `sls -2K` Command

```
# sls -2K aaa
-rw-rw----   1 root     other     20971704 Jun 15 17:12 aaa
---------- ----- sI {3,0,0,0}
-rw-rw----   1 root     other     10485760 Jun 15 17:12 aaa/1
---------- ----- sS
-rw-rw----   1 root     other     10485760 Jun 15 17:12 aaa/2
---------- ----- sS
-rw-rw----   1 root     other          184 Jun 15 17:12 aaa/3
---------- ----- sS
```

Restoring a Volume Overflow File Using Information From an Archiver Log

A volume overflow file is a file that is written on multiple volumes. If an archiver log file exists, you can search the archiver log for entries for the missing file. (See "To Set Up Archiver Logging" on page 61, if needed.) If you can find entries for a missing volume overflow file in an archiver log, you can use the file's position, segment size, VSN, and media type, to restore and reassemble the file using the `request`(1M), `star`(1M), `dd`(1M), and `cat`(1) commands. The procedure is described in "To Restore a Volume Overflow File Using Information From an Archiver Log" on page 88.

If needed, see TABLE 5-3 for definitions of the fields in the archiver log file.

The volume overflow file named `file3` is used in this section and in the procedure. CODE EXAMPLE 5-26 shows two entries for the two sections of file `file3` in the `archiver.log` file:

CODE EXAMPLE 5-26 Archiver Log File Entries

```
A 2004/08/23 10:28:51 sg 700036 ReleasePercent.1 12d55.1 qfs2
11731.1 89128448 ReleasePercent/huge2/dir24/file3 f 0 210
A 2004/08/23 10:28:51 sg 700034 ReleasePercent.1 15f9e.0 qfs2
11731.1 525271552 ReleasePercent/huge2/dir24/file3 f 1 220
```

The file is identified as a volume overflow file with two sections because the `f` in the third-to-last field indicates that the entry is for a regular file, and the `0` and the `1` in the second-to-last fields are section numbers. The fifth field shows that the file starts on VSN `700036` and overflows to information about `700034`.

The following procedure assumes that free space is available in the file system equal to two times the recovered file.

▼ To Restore a Volume Overflow File Using Information From an Archiver Log

---

Note – Free space must be available in the file system equal to two times the size of the file to be recovered.

---

1. **Use `vi`(1M) or another command to examine the archiver log file that contains an entry for the file you are trying to recover.**

   The archiver log file for `file3` was shown previously in CODE EXAMPLE 5-26.

2. **Use the `request`(1M) command to create a removable media file that points to each section.**

   CODE EXAMPLE 5-27 shows the `request`(1M) options to use.

   CODE EXAMPLE 5-27 Using `request`(1M) Commands

   ```
   # request -p 0x12d55 -m sg -v 700036 /samfs1/tp1
   # request -p 0x15f9e -m sg -v 700032 /samfs1/tp2
   ```

3. **Use the `cd`(1M) and `dd`(1M) commands to recover the sections.**

   CODE EXAMPLE 5-28 shows that a block size of 256 kilobytes is assumed for both tapes.

   CODE EXAMPLE 5-28 Using the `cd`(1) and `dd`(1M) Commands

   ```
   # cd /qfs2
   # dd  if=/samfs1/tp1 of=file3.0 ibs=256k
   340+0 records in
   174080+0 records out
   # dd  if=/samfs1/tp2 of=file3.1 ibs=256k
   2004+0 records in
   1026048+0 records out
   ```

   Repeat the `dd`(1M) command for each remaining section.

4. **Use the `ls`(1M) command to examine the output and ensure that all pieces of the file are on the disk.**

   CODE EXAMPLE 5-29 shows an example of this.

   CODE EXAMPLE 5-29 Using the `ls`(1) Command to Examine Output

   ```
   # ls -l file3.*
   -rw-r--r--   1 root     other   89128960  Aug 31 12:07 file3.0
   -rw-r--r--   1 root     other   525336576 Aug 31 12:14 file3.1
   ```

5. **Use the `cat`(1M) and `star`(1M) commands to reassemble the files.**

   CODE EXAMPLE 5-30 uses the `cat`(1) and `star`(1) commands to reassemble the files.

CODE EXAMPLE 5-30 Reassembling the Files

```
# cat file3.0 file3.1 > file3.2
# ls -l file3.*
-rw-r--r--   1 root     other    89128960  Aug 31 12:07 file3.0
-rw-r--r--   1 root     other    525336576 Aug 31 12:14 file3.1
-rw-r--r--   1 root     other    614465536 Aug 31 12:21 file3.2
# star xvbf 256 file3.2
ReleasePercent/huge2/dir24/file3
# sls -D ReleasePercent/huge2/dir24/file3
ReleasePercent/huge2/dir24/file3:
  mode: -rw-r--r--  links:   1  owner: root      group: other
  length: 614400000  admin id:      0  inode:    12481.1
  access:      Aug 31 12:40  modification: Aug 20 14:28
  changed:     Aug 31 12:43  attributes:   Aug 31 12:40
  creation:    Aug 31 12:40  residence:    Aug 31 12:40
```

Restoring Files Archived to Disk

The following sections describe how to restore files archived to disk. The sections are as follows:

- "To Gather Information for a Disk Archive Restoration" on page 91. This procedure shows you how to gather the information necessary to restore files from a disk archive `tar`(1) file. You need to perform this procedure before you perform the actual file restoration procedure.
- "To Restore a Single File From a Disk Archive `tar`(1) File" on page 96.
- "To Restore Multiple Files From a Disk Archive `tar`(1) File" on page 97.

The examples in these procedures uses six archived files. Three of these files use long path names (`filex`, `filey`, `filez`), which result in extended `tar`(1) file headers. These are included just for completeness. No extra steps are required on behalf of the user for disaster recovery purposes of files with long path names.

Each of the sample files has four archive copies, as follows:

- Copy 1 uses disk archive VSN `DISK_01`.
- Copy 2 uses disk archive VSN `DISK_02`.
- Copy 3 uses disk archive VSN `DISK_03`.

- Copy 4 uses tape archive VSN `000064`.

▼ To Gather Information for a Disk Archive Restoration

The following information is needed before you can restore any files that have been archived to disk:

- The disk volume name.
- The path to the disk archive `tar`(1) file.
- The path name defined for the disk volume name in `diskvols.conf`(4).

You must have either the saved `sls`(1) output or the archiver log files that include the files you want to restore.

1. **Find the disk volume name and the path to the disk archive `tar`(1) file that contains the archived file.**

   You can use either the `sls`(1) command with the `-D` option or the archiver log file entry. The examples in this step show both.

Method 1 - Using `sls`(1)

You can use this method if you have `sls`(1) output for the files you want to restore.

CODE EXAMPLE 5-31 shows output from the `sls`(1) command. Examine the lines that correspond to the disk archive copies. These lines show a media type of `dk` in the fifth field. The second-to-the-last field in these lines shows the disk volume name as defined in `diskvols.conf`(4). The last field shows the path to the disk archive `tar`(1) file.

In the output for CODE EXAMPLE 5-31, both the commands you type in and the information you are looking for are in **bold** text.

CODE EXAMPLE 5-31 Sample `sls`(1) Output for a File Archived to Disk

```
# cd /shareqfs2/testdir4
# sls -D filea fileb filec
filea:
  mode: -rw-r--r--  links:   1  owner: root      group: other
  length:     65732  admin id:      0  inode:   120235.783
  archdone;
  copy 1: ---- Nov  3 14:46     81366.1    dk DISK_01 d8/d19/f102
  copy 2: ---- Nov  3 14:54     2ec7e.209  dk DISK_02 d2/d236/f126
  copy 3: ---- Nov  3 14:58        bf.209  dk DISK_03 f191
```

CODE EXAMPLE 5-31 Sample `sls`(1) Output for a File Archived to Disk *(Continued)*

```
  copy 4: ---- Nov  3 15:05      ea7a.209  lt 000064
  access:      Nov  3 14:35  modification: Nov  3 14:35
  changed:     Nov  3 14:35  attributes:   Nov  3 14:35
  creation:    Nov  3 14:35  residence:    Nov  3 14:35
fileb:
  mode: -rw-r--r--  links:   1  owner: root      group: other
  length:     65732  admin id:      0  inode:   120300.783
  archdone;
  copy 1: ---- Nov  3 14:46     81366.105  dk DISK_01 d8/d19/f102
  copy 2: ---- Nov  3 14:54     2ec7e.411  dk DISK_02 d2/d236/f126
  copy 3: ---- Nov  3 14:58        bf.411  dk DISK_03 f191
  copy 4: ---- Nov  3 15:05      ea7a.411  lt 000064
  access:      Nov  3 14:35  modification: Nov  3 14:35
  changed:     Nov  3 14:35  attributes:   Nov  3 14:35
  creation:    Nov  3 14:35  residence:    Nov  3 14:35
filec:
  mode: -rw-r--r--  links:   1  owner: root      group: other
  length:     65732  admin id:      0  inode:   120243.783
  archdone;
  copy 1: ---- Nov  3 14:46     81366.83   dk DISK_01 d8/d19/f102
  copy 2: ---- Nov  3 14:54     2ec7e.38f  dk DISK_02 d2/d236/f126
  copy 3: ---- Nov  3 14:58        bf.38f  dk DISK_03 f191
  copy 4: ---- Nov  3 15:05      ea7a.38f  lt 000064
  access:      Nov  3 14:35  modification: Nov  3 14:35
  changed:     Nov  3 14:35  attributes:   Nov  3 14:35
  creation:    Nov  3 14:35  residence:    Nov  3 14:35
# cd /shareqfs2/testdir4/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0
001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir
0001/tstdir0001/tstdir0001
# sls -D filex filey filez
filex:
  mode: -rw-r--r--  links:   1  owner: root      group: other
  length:    131420  admin id:      0  inode:   120239.783
  archdone;
  copy 1: ---- Nov  3 14:50     81367.20b  dk DISK_01 d8/d19/f103
  copy 2: ---- Nov  3 14:54     2ec7e.28d  dk DISK_02 d2/d236/f126
  copy 3: ---- Nov  3 14:58        bf.28d  dk DISK_03 f191
  copy 4: ---- Nov  3 15:05      ea7a.28d  lt 000064
  access:      Nov  3 14:36  modification: Nov  3 14:36
  changed:     Nov  3 14:36  attributes:   Nov  3 14:36
  creation:    Nov  3 14:36  residence:    Nov  3 14:36
filey:
  mode: -rw-r--r--  links:   1  owner: root      group: other
  length:    131420  admin id:      0  inode:   120232.783
  archdone;
  copy 1: ---- Nov  3 14:50     81367.107  dk DISK_01 d8/d19/f103
  copy 2: ---- Nov  3 14:54     2ec7e.107  dk DISK_02 d2/d236/f126
```

CODE EXAMPLE 5-31 Sample `sls`(1) Output for a File Archived to Disk *(Continued)*

```
  copy 3: ---- Nov  3 14:58        bf.107  dk DISK_03 f191
  copy 4: ---- Nov  3 15:05      ea7a.107  lt 000064
  access:      Nov  3 14:36  modification: Nov  3 14:36
  changed:     Nov  3 14:36  attributes:   Nov  3 14:36
  creation:    Nov  3 14:36  residence:    Nov  3 14:36
filez:
  mode: -rw-r--r--  links:   1  owner: root      group: other
  length:    131420  admin id:      0  inode:   120228.783
  archdone;
  copy 1: ---- Nov  3 14:50     81367.3    dk DISK_01 d8/d19/f103
  copy 2: ---- Nov  3 14:54     2ec7e.3    dk DISK_02 d2/d236/f126
  copy 3: ---- Nov  3 14:58        bf.3    dk DISK_03 f191
  copy 4: ---- Nov  3 15:05      ea7a.3    lt 000064
  access:      Nov  3 14:36  modification: Nov  3 14:36
  changed:     Nov  3 14:36  attributes:   Nov  3 14:36
  creation:    Nov  3 14:36  residence:    Nov  3 14:36
```

Method 2 - Using the Archiver Log File

Use this method if you have an archiver log file that includes the files you want to restore.

CODE EXAMPLE 5-32 shows archiver log example output. Examine the lines corresponding to the disk archive copies. These lines show a media type of `dk` in the fourth field. The fifth field in these lines shows the disk volume name as defined in `diskvols.conf`(4), and it shows the path to the disk archive `tar`(1) file. These are joined by a slash (`/`) character.

In the output for CODE EXAMPLE 5-32, the information you are looking for is in **bold** text.

CODE EXAMPLE 5-32 Archiver Log File Output for a File Archived to Disk

```
A 2003/11/03 14:46:35 dk DISK_01/d8/d19/f102 arset4.1 81366.1 shareqfs2
120235.783 65732 testdir4/filea f 0 0
A 2003/11/03 14:46:35 dk DISK_01/d8/d19/f102 arset4.1 81366.83 shareqfs2
120243.783 65732 testdir4/filec f 0 0
A 2003/11/03 14:46:35 dk DISK_01/d8/d19/f102 arset4.1 81366.105 shareqfs2
120300.783 65732 testdir4/fileb f 0 0
A 2003/11/03 14:50:35 dk DISK_01/d8/d19/f103 arset4.1 81367.3 shareqfs2
120228.783 131420
testdir4/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tst
dir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstd
ir0001/filez f 0 0
A 2003/11/03 14:50:35 dk DISK_01/d8/d19/f103 arset4.1 81367.107 shareqfs2
120232.783 131420
```

CODE EXAMPLE 5-32 Archiver Log File Output for a File Archived to Disk *(Continued)*

```
testdir4/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tst
dir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstd
ir0001/filey f 0 0
A 2003/11/03 14:50:35 dk DISK_01/d8/d19/f103 arset4.1 81367.20b shareqfs2
120239.783 131420
testdir4/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tst
dir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstd
ir0001/filex f 0 0
A 2003/11/03 14:54:35 dk DISK_02/d2/d236/f126 arset4.2 2ec7e.3 shareqfs2
120228.783 131420
testdir4/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tst
dir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstd
ir0001/filez f 0 0
A 2003/11/03 14:54:35 dk DISK_02/d2/d236/f126 arset4.2 2ec7e.107 shareqfs2
120232.783 131420
testdir4/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tst
dir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstd
ir0001/filey f 0 0
A 2003/11/03 14:54:35 dk DISK_02/d2/d236/f126 arset4.2 2ec7e.209 shareqfs2
120235.783 65732 testdir4/filea f 0 0
A 2003/11/03 14:54:35 dk DISK_02/d2/d236/f126 arset4.2 2ec7e.28d shareqfs2
120239.783 131420
testdir4/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tst
dir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstd
ir0001/filex f 0 0
A 2003/11/03 14:54:35 dk DISK_02/d2/d236/f126 arset4.2 2ec7e.38f shareqfs2
120243.783 65732 testdir4/filec f 0 0
A 2003/11/03 14:54:35 dk DISK_02/d2/d236/f126 arset4.2 2ec7e.411 shareqfs2
120300.783 65732 testdir4/fileb f 0 0
A 2003/11/03 14:58:35 dk DISK_03/f191 arset4.3 bf.3 shareqfs2 120228.783 131420
testdir4/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tst
dir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstd
ir0001/filez f 0 0
A 2003/11/03 14:58:35 dk DISK_03/f191 arset4.3 bf.107 shareqfs2 120232.783
131420
testdir4/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tst
dir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstd
ir0001/filey f 0 0
A 2003/11/03 14:58:35 dk DISK_03/f191 arset4.3 bf.209 shareqfs2 120235.783 65732
testdir4/filea f 0 0
A 2003/11/03 14:58:35 dk DISK_03/f191 arset4.3 bf.28d shareqfs2 120239.783
131420
testdir4/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tst
dir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstd
ir0001/filex f 0 0
A 2003/11/03 14:58:35 dk DISK_03/f191 arset4.3 bf.38f shareqfs2 120243.783 65732
testdir4/filec f 0 0
```

CODE EXAMPLE 5-32 Archiver Log File Output for a File Archived to Disk *(Continued)*

```
A 2003/11/03 14:58:35 dk DISK_03/f191 arset4.3 bf.411 shareqfs2 120300.783 65732
testdir4/fileb f 0 0
A 2003/11/03 15:05:33 lt 000064 arset4.4 ea7a.3 shareqfs2 120228.783 131420
testdir4/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tst
dir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstd
ir0001/filez f 0 42
A 2003/11/03 15:05:33 lt 000064 arset4.4 ea7a.107 shareqfs2 120232.783 131420
testdir4/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tst
dir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstd
ir0001/filey f 0 42
A 2003/11/03 15:05:33 lt 000064 arset4.4 ea7a.209 shareqfs2 120235.783 65732
testdir4/filea f 0 42
A 2003/11/03 15:05:33 lt 000064 arset4.4 ea7a.28d shareqfs2 120239.783 131420
testdir4/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tst
dir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstdir0001/tstd
ir0001/filex f 0 42
A 2003/11/03 15:05:33 lt 000064 arset4.4 ea7a.38f shareqfs2 120243.783 65732
testdir4/filec f 0 42
A 2003/11/03 15:05:33 lt 000064 arset4.4 ea7a.411 shareqfs2 120300.783 65732
testdir4/fileb f 0 42
```

2. **Use** `cat`**(1) or another command to examine the** `diskvols.conf`**(4) file and to find the path name defined for the disk volume name in** `diskvols.conf`**(4).**

   CODE EXAMPLE 5-33 shows the following three disk volumes defined for receiving disk archive copies:

   - Disk volume name `DISK_01` points to the destination path `/ufs2/disk_archive/01` that is mounted locally.
   - Disk volume name `DISK_02` points to the destination path `/ufs2/disk_archive/02` that is mounted locally.
   - Disk volume name `DISK_03` points to the destination path `/qfs1/disk_archive/03` on remote server `mars`.

   CODE EXAMPLE 5-33 Example `diskvols.conf`(4) Entries

   ```
   # cat /etc/opt/SUNWsamfs/diskvols.conf
   DISK_01 /ufs2/disk_archive/01
   DISK_02 /ufs2/disk_archive/02
   DISK_03 mars:/qfs1/disk_archive/03
   ```

To Restore a Single File From a Disk Archive `tar`(1) File

Use this procedure to restore a single file from a disk archive `tar`(1) file.

1. **Use the procedure in "To Gather Information for a Disk Archive Restoration" on page 91 to gather the information needed for this restoration.**

2. **Use the** `mkdir`**(1) command to create a directory in which to restore the files within a SAM-QFS file system.**

3. **Use the** `cd`**(1) command to change to the restoration directory.**

4. **Use the** `star`**(1M) command with its** `-tv` **options to list the content of the disk archive** `tar`**(1) file.**

   CODE EXAMPLE 5-34 shows the content of the disk archive `tar`(1) file associated with archive copy 1. The `star`(1M) command shown derives its input file name from two sources:

   - The `/ufs2/disk_archive/01` is derived from the `diskvols.conf`(4) file.
   - The `/d8/d19/f102` is derived from the `sls`(1) output or the archiver log file.

   CODE EXAMPLE 5-34 Listing the Content of the `tar`(1) File

```
# star -tv -f /ufs2/disk_archive/01/d8/d19/f102
-rw-r--r-- root/other    65732 2003-11-03 14:35 testdir4/filea
-rw-r--r-- root/other    65732 2003-11-03 14:35 testdir4/filec
-rw-r--r-- root/other    65732 2003-11-03 14:35 testdir4/fileb
```

Note – This tar file could be on a remote server. If that is the case, this step requires that the remote authentication database be configured properly in order to access the disk archive `tar`(1) file on the remote server. For information on configuring the `/.rhosts` file, see the `hosts.equiv`(4) man page.

---

5. **Verify that the file you wish to restore is listed in the output from Step 4.**

   If you find the file you are looking for, copy down the exact path name shown in the output for use in Step 6.

   CODE EXAMPLE 5-34 shows lost file `fileb` residing in directory `testdir4`. Copy down `testdir4/fileb` for use in Step 6.

6. **Use the** `star`**(1M) command with its** `-xv` **options to restore the file to the current directory.**

   The file name must match exactly the path name as it was listed in the output from Step 4 (CODE EXAMPLE 5-34).

CODE EXAMPLE 5-35 shows the `star`(1M) command retrieving file `testdir4/fileb` from disk archive `tar`(1) file `/ufs2/disk_archive/01/d8/d19/f102`.

CODE EXAMPLE 5-35 Using `star`(1M) to Retrieve the File

```
# star -xv -f /ufs2/disk_archive/01/d8/d19/f102 testdir4/fileb
testdir4/fileb
```

7. **Use the `sls`(1) command with its `-DR` options to verify that you have extracted the proper file.**

   CODE EXAMPLE 5-36 shows `sls`(1) output.

CODE EXAMPLE 5-36 `sls`(1) Output Showing the Retrieved Files

```
# sls -DR
testdir4:
  mode: drwxr-xr-x  links:   2  owner: root      group: other
  length:      4096  admin id:      0  inode:   120274.787
  access:      Nov  4 14:08  modification: Nov  4 14:08
  changed:     Nov  4 14:08  attributes:   Nov  4 14:08
  creation:    Nov  4 14:08  residence:    Nov  4 14:08


testdir4:
testdir4/fileb:
  mode: -rw-r--r--  links:   1  owner: root      group: other
  length:     65732  admin id:      0  inode:   120293.785
  access:      Nov  4 14:08  modification: Nov  3 14:35
  changed:     Nov  4 14:08  attributes:   Nov  4 14:08
  creation:    Nov  4 14:08  residence:    Nov  4 14:08
```

▼ To Restore Multiple Files From a Disk Archive `tar`(1) File

Use this procedure to restore multiple files from a disk archive `tar`(1) file.

1. **Use the procedure in "To Gather Information for a Disk Archive Restoration" on page 91 to gather the information needed for this restoration.**
2. **Use the `mkdir`(1) command to create a directory in which to restore the files within a SAM-QFS file system.**
3. **Use the `cd`(1) command to change to the restoration directory.**

4. **Use the `star`(1M) command with its `-tv` options to list the content of the disk archive `tar`(1) file.**

   CODE EXAMPLE 5-37 shows the content of the disk archive `tar`(1) file associated with archive copy 1. The `star`(1M) command shown derives its input file name from two sources:

   - The `/ufs2/disk_archive/01` is derived from the `diskvols.conf`(4) file.
   - The `/d8/d19/f102` is derived from the `sls`(1) output or the archiver log file.

CODE EXAMPLE 5-37 Listing the Content of the `tar`(1) File

```
# star -tv -f /ufs2/disk_archive/01/d8/d19/f102
-rw-r--r-- root/other    65732 2003-11-03 14:35 testdir4/filea
-rw-r--r-- root/other    65732 2003-11-03 14:35 testdir4/filec
-rw-r--r-- root/other    65732 2003-11-03 14:35 testdir4/fileb
```

Note – This tar file could be on a remote server. If that is the case, this step requires that the remote authentication database be configured properly in order to access the disk archive `tar`(1) file on the remote server. For information on configuring the `/.rhosts` file, see the `hosts.equiv`(4) man page.

---

5. **Verify that the files you wish to restore are listed in the output from Step 4.**

6. **Use the `star`(1M) command with its `-xv` options to restore the entire content of the disk archive `tar`(1) file to the current directory.**

   CODE EXAMPLE 5-38 shows the `star`(1M) command retrieving all files from disk archive `tar`(1) file `/ufs2/disk_archive/01/d8/d19/f102`.

CODE EXAMPLE 5-38 Using `star`(1M) to Retrieve all the Files

```
# star -xv -f /ufs2/disk_archive/01/d8/d19/f102
testdir4/filea
testdir4/filec
testdir4/fileb
```

7. **Use the `sls`(1) command with its `-DR` options to verify that you have extracted the proper files.**

   CODE EXAMPLE 5-39 shows `sls`(1) output.

CODE EXAMPLE 5-39 `sls`(1) Output Showing the Retrieved Files

```
# sls -DR
testdir4:
  mode: drwxr-xr-x  links:   2  owner: root      group: other
```

CODE EXAMPLE 5-39 `sls`(1) Output Showing the Retrieved Files *(Continued)*

```
    length:       4096  admin id:      0  inode:   120274.789
    access:       Nov  4 14:11  modification: Nov  4 14:11
    changed:      Nov  4 14:11  attributes:   Nov  4 14:11
    creation:     Nov  4 14:11  residence:    Nov  4 14:11

  testdir4:
  testdir4/filea:
    mode: -rw-r--r--  links:   1  owner: root      group: other
    length:      65732  admin id:      0  inode:   120293.787
    access:       Nov  4 14:11  modification: Nov  3 14:35
    changed:      Nov  4 14:11  attributes:   Nov  4 14:11
    creation:     Nov  4 14:11  residence:    Nov  4 14:11

  testdir4/fileb:
    mode: -rw-r--r--  links:   1  owner: root      group: other
    length:      65732  admin id:      0  inode:   120281.783
    access:       Nov  4 14:11  modification: Nov  3 14:35
    changed:      Nov  4 14:11  attributes:   Nov  4 14:11
    creation:     Nov  4 14:11  residence:    Nov  4 14:11

  testdir4/filec:
    mode: -rw-r--r--  links:   1  owner: root      group: other
    length:      65732  admin id:      0  inode:   120280.783
    access:       Nov  4 14:11  modification: Nov  3 14:35
    changed:      Nov  4 14:11  attributes:   Nov  4 14:11
    creation:     Nov  4 14:11  residence:    Nov  4 14:11
```

Retrieving Unarchived Files From File Systems

Unarchived files that resided within a SAM-QFS file system might not be recoverable after a system outage. The following list contains information that might help you to retrieve unarchived files:

- If the `samfsdump`(1M) method was used to dump and back up metadata, the `samfsrestore`(1M) command identifies files without archive copies and flags them as damaged.
- Sun StorEdge SAM-FS log files cannot help you determine which files were not archived and were therefore lost between the last archiver run and the system outage. However, you can determine the files that might not have been archived

by analyzing the `archiver.cmd` file for archiving directives and intervals. If all files are eligible for archiving, you can find the age of the oldest unarchived (lost) files in the `archiver.cmd` file's contents.

- You can use the `-l` and `-v` options with the `archiver`(1M) command to generate information you can use to determine whether volumes were available to archive each archive set's data before the outage. Lack of sufficient volumes can prevent archiving of data in one or more archive sets. For information about the `archiver`(1M) command, see the `sam-archiverd`(1M) man page.
- If you are recovering files straight from a backup tape in `tar`(1) format, the files are restored to their locations according to the information on the tape. The path name is relative to the mount point of the file system. If any files have been moved within the system since the archive copies were created, they are restored to their original locations, not to their new locations.
- You can use the `sfind`(1M) command line to identify all files in a file system that are not archived. The following command finds all unarchived files associated with the `/sam1` mount point:

```
# sfind /sam1 ! -archived
```

CHAPTER 6

Salvaging Damaged Volumes

This chapter describes how to restore data from tapes or magneto-optical disks that are not usable in a SAM-QFS environment. This procedures in this chapter describe what to do when a volume is partially corrupted, accidentally relabeled, has a destroyed label, or is entirely destroyed. The procedures in this chapter describe how to recover data both when archive copies are available and when there are no other copies available.

Before attempting the procedures in this chapter, determine whether or not the volume can be read by using software other than Sun StorEdge SAM-FS tools. Try reading the volume in multiple drives, or try using the `tar`(1) command.

This chapter covers the following topics:

Recovering Data From a Tape Volume

The procedures for recovering data from a tape volume differ depending on the nature of the damage and whether or not additional archive copies of the volume's files are present on another tape. This section describes how to recover data in the following scenarios:

- Tape volume is damaged, and alternative archive copies are available.
- Tape volume is partially corrupt, and no alternative archive copies are available.
- Tape volume is accidentally relabeled, and no alternative archive copies are. available.
- The Sun StorEdge SAM-FS software cannot read the tape volume label and no alternative archive copies are available.

Damaged Tape Volume – Other Copies Available

The Sun StorEdge SAM-FS storage and archive manager allows you to make up to four archive copies of each online file. By default, only one copy is made, but Sun Microsystems recommends that you make at least two copies, preferably to physically different archive media.

When an alternative archive copy is available, the recovery procedure includes a step for rearchiving all archive copies currently stored on the damaged volume before dispensing with the damaged volume. The new archive copies are made from the available alternative archive copy.

▼ To Recycle a Damaged Tape – Other Copies Available

Use this procedure if alternative archive copies exist on volumes that are stored on-site and are available for staging.

1. **Export the damaged volume from the tape library, and flag it as unavailable in the historian catalog.**

   Enter the `export`(1M) and `chmed`(1M) commands as shown in the following screen example, specifying the media type (*mt*) and VSN (*vsn*) of the damaged volume.

```
# export mt.vsn
# chmed +U mt.vsn
```

2. **Flag the unavailable volume for recycling.**

   Use the `chmed`(1M) command and specify the media type (*mt*) and the VSN (*vsn*) of the damaged volume.

```
# chmed +c mt.vsn
```

3. **Set the `-ignore` option for the library in the `recycler.cmd` file.**

   CODE EXAMPLE 6-1 shows the `-ignore` option set on the `lt20` library:

CODE EXAMPLE 6-1 Example `recycler.cmd` File with the `-ignore` Option

```
# vi /etc/opt/SUNWsamfs/recycler.cmd
logfile = /var/adm/recycler.log
lt20 -hwm 75 -mingain 60 -ignore
:wq
```

For more information about the `ignore` option, see the `recycler-cmd`(4) man page.

4. **Run the `sam-recycler`(1M) command with the `-x` option from the command line.**

   For example:

   ```
   # sam-recycler -x
   ```

   When the recycler runs, it does not select any volumes for recycling other than the volume you have marked as unavailable. The recycler identifies all active archive copies on this volume and flags those archive copies for rearchiving. The next time the archiver runs, the archive copies marked for rearchiving will be written to new volumes.

   After the archive copies have been written to new volumes, the damaged volume you are recycling is considered to be drained of active archive copies.

5. **Dispense with the volume.**

   After the damaged volume is drained of active archive copies, you can dispense with the volume. How you dispense with it depends on the nature of the damage. Use the following guidelines:

   - If the tape was accidentally relabeled, use the `tplabel`(1M) command to relabel the volume.
   - If the tape label is unreadable, use the `tplabel`(1M) command to relabel the volume.
   - If relabeling the volume fails, export the volume from the historian and dispose of the tape.

   If the tape is either partially corrupt or completely destroyed, it is possible (but not recommended) to reuse the tape VSN after the volume has been exported from the historian catalog.

Damaged Tape Volume – No Other Copies Available

If a tape volume is partially corrupt, it is possible to recover data from the parts of the tape volume that are not corrupt. This process is not an exact science, and it requires some trial and error to recover as much data as possible.

Errors logged in the device log can help you determine the area of a tape that is damaged. The `archive_audit`(1M) command can be used to generate the position and offset information for all archived files for a specific file system. You can use this position and offset information to help determine which archive copies are written to an area of a tape that is damaged.

▼ To Recover Files From a Damaged Tape – No Other Copies Available

1. **Use the `archive_audit`(1M) command to generate a list of all files with archive copies on the partially corrupt tape volume.**

   Use the command syntax shown in the following screen example, specifying the file system's mount point, the VSN (*vsn*) of the volume, and an output file name.

```
# archive_audit /mount_point | grep vsn > filename
```

2. **Edit the output file from the `archive_audit`(1M) command in the previous step, deleting the lines for the files in the damaged area, and saving the list of deleted files for inspection in Step 3.**

3. **Use the list of files with archive copies that cannot be accessed (the ones that are written in the area of the tape determined to be damaged) to determine if any of the files are still on the disk.**

   Files that are not on disk cannot be recovered. These unrecoverable files can be removed from the file system.

4. **Edit and run the `stageback.sh` script on the `archive_audit` output file you edited in Step 2.**

   The `stageback.sh` script can stage each file from `archive_audit` output, set it to `no-release`, and mark the file for rearchiving.

   For information about the `stageback.sh` script, see "Disaster Recovery Commands and Tools" on page 53.

   a. **Open the `/opt/SUNWsamfs/examples/stageback.sh` file for editing.**

```
# cd /opt/SUNWsamfs/examples
# vi stageback.sh
```

   b. **Find the section that begins with `# echo rearch $file`.**

   CODE EXAMPLE 6-2 shows this.

CODE EXAMPLE 6-2 Example `stageback.sh` File

```
# echo rearch $file
#
# Edit the following line for the correct media type and VSN
#
# eval /opt/SUNWsamfs/bin/rearch –m media –v VSN $file
```

c. **In the section shown in** CODE EXAMPLE 6-2**, replace the word** `media` **with the media type (*mt*) and the word** `VSN` **with the VSN of the damaged volume, which are the same as the VSNs in Step 1.**

d. **Remove the pound sign from the beginning of the lines in the section shown in Step b.**

   The file should now look like CODE EXAMPLE 6-3.

CODE EXAMPLE 6-3 Example `stageback.sh` File - Edited

```
echo rearch $file

# Edit the following line for the correct media type and VSN

eval /opt/SUNWsamfs/bin/rearch -m media -v VSN $file
```

e. **Save and quit the file.**

f. **Run the** `stageback.sh` **script.**

Relabeled Tape Volume – No Other Copies Available

The Sun StorEdge SAM-FS software cannot read beyond the EOD. If a tape is accidentally relabeled, the only possibility for recovering any data is to contact the tape manufacturer to determine if they offer a method for reading beyond EOD.

If the tape manufacturer can provide a mechanism for reading beyond EOD, you can recover the data by combining that process with the procedure for recovering files from a tape volume with a label not readable by the Sun StorEdge SAM-FS software. This procedure is described under "Unreadable Tape Label – No Other Copies Available" on page 105.

Unreadable Tape Label – No Other Copies Available

Whenever the Sun StorEdge SAM-FS software receives a request to mount a tape volume into a drive, one of the first actions taken is to verify the tape label written on the tape. If the tape label cannot be read, the Sun StorEdge SAM-FS software cannot use the tape for staging or archiving activities.

The `tarback.sh`(1M) script is used to recover data from a tape that has a label that cannot be read. The shell script automates the process of recovering data written to a tape, using the `star`(1M) command to read each archive file written on a specific tape volume. The file data is read back onto disk (into a Sun StorEdge QFS or UFS file system) as data. File data recovered in this manner can then be moved to the appropriate location in the Sun StorEdge QFS file system. It must then be archived as new data.

▼ To Recover Files From a Tape Whose Label Is Unreadable

1. **If you are using this process to recover file data from several tapes, disable any currently occurring recycling.**

   When recycling is going on, data on the tape volumes may be inaccessible.

2. **Use the `cp`(1M) command to copy the `tarback.sh` file to a working location.**

   For example, the following command copies the script from the default location `/opt/SUNWsamfs/examples/tarback.sh` to `/var/tarback.sh`.

```
# cp /opt/SUNWsamfs/examples/tarback.sh /var/tarback.sh
```

3. **Enter the `samcmd`(1M) command with the `unavail` option to make the tape drive unavailable.**

   To prevent the tape drive from being used for staging and archiving activities, use the syntax shown in the following screen example. Specify the `Equipment Ordinal` of the drive, as specified in the `mcf`(4) file, for *eq*.

```
# samcmd unavail eq
```

4. **Edit the working copy of the `tarback.sh`(1M) script to specify the variables shown in the following table.**

TABLE 6-1 Variables to Specify in the `tarback.sh`(1M) Script

Variable	**Definition**
`EQ="`*eq*`"`	The Equipment Ordinal of the tape drive as defined in the `mcf` file.
`TAPEDRIVE="`*path*`"`	The raw path to the device that is described by `EQ=`.

TABLE 6-1 Variables to Specify in the `tarback.sh`(1M) Script *(Continued)*

Variable	Definition
`BLOCKSIZE="`*size*`"`	The block size in 512-byte units. Specify `256` for a block size of `128` kilobytes.
`MEDIATYPE="`*mt*`"`	The two-character media type for this tape as defined in the `mcf`(4) man page.

`VSN_LIST="`*vsn1 vsn2 ...*`"`	The list of VSNs to be read. There is no limit on the number of VSNs that can be specified. Use a space character to separate the VSNs.
	This list can be continued onto another line by using a backslash (\) character. For example:
	`VSN_LIST="vsn1 vsn2 \`
	`vsn3"`

5. **Execute the** `tarback.sh`**(1M) script.**

Recovering Data From a Magneto-optical Volume

The procedures for recovering data from a magneto-optical volume differ, depending on the nature of the damage and whether or not additional archive copies of the volume's files are present on another tape. This section describes how to recover data in the following scenarios:

- Magneto-optical volume is damaged, and alternative archive copies are available.

  See "Damaged Magneto-optical Volume – Copies Available" on page 108.
- Magneto-optical volume is damaged, and no alternative archive copies are available.

  See "Damaged Magneto-optical Volume – No Other Copies Available" on page 110.
- Magneto-optical volume is accidentally relabeled, and no alternative archive copies are available.

  See "Relabeled Magneto-optical Volume – No Other Copies Available" on page 112.
- The Sun StorEdge SAM-FS software cannot read the magneto-optical volume label, and no alternative archive copies are available.

  See "Unreadable Label – No Other Copies Available" on page 112.

Damaged Magneto-optical Volume – Copies Available

Regardless of the nature of the damage to the magneto-optical volume, if an alternative archive copy is available, you should use the good magneto-optical volume as your primary set of archive copies.

The recovery procedure includes a step for rearchiving all archive copies currently stored on the damaged volume before dispensing with the damaged volume. The new archive copies are made from the available alternative archive copy.

▼ To Rearchive Files and Recycle a Damaged Magneto-optical Volume—Copies Available

Use this procedure if readable alternative archive copies exist on volumes that are available on-site for staging.

1. **Enter the `samexport`(1M) command to export the damaged volume from the magneto-optical library.**

   Use the syntax shown in the following screen example, specifying the media type (*mt*) and VSN (*vsn*) of the damaged volume.

```
# samexport mt.vsn
```

2. **Enter the `chmed`(1M) command with the `-U` option to flag the damaged volume as unavailable in the historian catalog.**

   Use the syntax shown in the following screen example, specifying the media type (*mt*) and VSN (*vsn*) of the damaged volume.

```
# chmed +U mt.vsn
```

3. **Enter the `chmed`(1M) command with the `-c` option to flag the unavailable volume for recycling.**

   Use the syntax shown in the following screen example, specifying the media type (*mt*) and the VSN (*vsn*) of the damaged volume.

```
# chmed +c mt.vsn
```

4. **Edit the `recycler.cmd`(4) file to set the `-ignore` option for the library.**

   The following screen example shows the `-ignore` option set on the `lt20` library.

   CODE EXAMPLE 6-4 Example `recycler.cmd` File with the `-ignore` Option

```
# vi /etc/opt/SUNWsamfs/recycler.cmd
logfile = /var/adm/recycler.log
lt20 -hwm 75 -mingain 60 -ignore
:wq
```

5. **Enter the `sam-recycler`(1M) command with the `-x` option.**

```
# sam-recycler -x
```

   When the recycler runs, it does not select any volumes for recycling other than the volume you have marked as unavailable. The recycler identifies all active archive copies on this volume and flags those archive copies for rearchiving. The next time the archiver runs, the archive copies marked for rearchiving are written to new volumes.

   After the archive copies have been written to new volumes, the damaged volume you are recycling is considered to be drained of active archive copies.

6. **Dispense with the volume.**

   After the damaged volume is drained for active archive copies, you can dispense with the volume. How you dispense with it depends on the nature of the damage. See the following guidelines:

   - If the magneto-optical volume was accidentally relabeled, use the `odlabel`(1M) command to relabel the volume.
   - If the magneto-optical label is unreadable, export the volume from the historian and dispose of the magneto-optical volume.
   - If the magneto-optical volume is partially corrupt, export the volume from the historian and dispose of the magneto-optical volume.
   - If the magneto-optical volume is completely destroyed, export the volume from the historian and dispose of the magneto-optical volume.

   If the magneto-optical volume is either partially corrupt or completely destroyed, it is possible (but not recommended) to reuse the magneto-optical label after the volume has been exported from the historian catalog.

   If the magneto-optical volume is completely destroyed and no alternative archive copies exist, there is no chance for recovering any data from this magneto-optical platter.

Damaged Magneto-optical Volume – No Other Copies Available

If a magneto-optical volume is only partially corrupt, it is possible to recover data written to the parts of the magneto-optical volume that are not damaged. This process requires some trial and error to recover as much data as possible.

It is possible to determine the area of a magnetic optical volume that is damaged from errors logged in the device logs. By using file names for files that cannot be retrieved, you can determine the location of the damage using the position and offset data.

The `archive_audit`(1M) command audits all archive copies for a specific file system. The output of the `archive_audit` command includes the position and offset information for each archive copy. You can use this position and offset information to help determine which archive copies are written to an area of a damaged magneto-optical disk.

▼ To Recover From a Damaged Magneto-optical Volume – No Other Copies Available

Copies of files that were archived outside the damaged area on a magneto-optical volume may be accessible. You can use the following procedure to recover files in accessible areas of a partially corrupted magneto-optical volume.

1. **Use the `archive_audit`(1M) command to generate a list of all files with archive copies on the partially corrupt tape volume:**

   Use the syntax shown in the following screen example, specifying the file system's mount point, the VSN of the damaged volume, and an output file name.

```
# archive_audit /mount_point | grep vsn > filename
```

2. **Edit the `archive_audit` output file and create three separate files with the following contents:**
   - Files that appear before the damaged area on the magneto-optical disk
   - Files that appear within the damaged area
   - Files that appear after the damaged area.

3. **Look for the files with archive copies within the damaged area of the magneto-optical disk to determine if any of the files are still in disk cache.**

   Files that are not in disk cache cannot be recovered.

4. **Remove unrecoverable files from Step 2 from the file system.**

5. **Edit and run the `stageback.sh` script using the files created in Step 2 that list files outside the damaged area.**

   The `stageback.sh` script stages each file from `archive_audit` output, sets it to `no-release`, and marks the file for rearchiving.

   For information about the `stageback.sh` script, see Chapter 1.

   a. **Open the `/opt/SUNWsamfs/examples/stageback.sh` file for editing.**

```
# cd /opt/SUNWsamfs/examples
# vi stageback.sh
```

   b. **Find the section that begins with `# echo rearch $file`.**

CODE EXAMPLE 6-5 Example `stageback.sh` File

```
# echo rearch $file
#
# Edit the following line for the correct media type and VSN
#
# eval /opt/SUNWsamfs/bin/rearch -m media -v VSN $file
```

   c. **In the section shown in** CODE EXAMPLE 6-5**, replace the word `media` with the media type and the word `VSN` with the same VSN specified in Step 1.**

   d. **Remove the pound sign from the beginning of the lines in the section shown in Step b.**

CODE EXAMPLE 6-6 Example `stageback.sh` File - Edited

```
echo rearch $file

# Edit the following line for the correct media type and VSN

eval /opt/SUNWsamfs/bin/rearch -m media -v VSN $file
```

   e. **Save and quit the file.**

   f. **Run the `stageback.sh` script.**

Relabeled Magneto-optical Volume – No Other Copies Available

Unlike tape media, magneto-optical media does not have an EOD marker. When a magneto-optical volume is accidentally relabeled, the Sun StorEdge SAM-FS software cannot access data written previously because of the label date. The Sun StorEdge SAM-FS systems assume that if the label date on the magneto-optical volume is newer than the archive copy date of files, that data is no longer accessible.

Contact Sun Microsystems customer support if a magneto-optical volume is accidentally relabeled. It is sometimes possible to recover some of this data with a special (but unsupported) `samst` driver that ignores the magneto-optical label date. This driver is not a standard part of the Sun StorEdge SAM-FS product, and it is not released as part of the product. It can only be made available by Sun's customer support.

Unreadable Label – No Other Copies Available

For magneto-optical media, there is no standard Solaris approach for locating and skipping to the various `tar`(1M) files. Contact Sun Microsystems customer support if you need to access files on a magneto-optical volume with an unreadable label.

CHAPTER 7

Recovering File Systems

This chapter describes how to recover data when a SAM-QFS file system is corrupted or lost. These procedures differ, depending on the type of file system and whether or not you have a `samfsdump`(1M) of the file system available. You might require the assistance from your ASP or a Sun Microsystems customer support staff member for this process to be successful.

This chapter covers the following topics

- "Recovering a SAM-QFS File System With a Metadata Dump File" on page 113
- "Recovering a SAM-QFS File System Without a Dump File" on page 115

Recovering a SAM-QFS File System With a Metadata Dump File

If you have `samfsdump`(1M) metadata output for a file system, you can use the `samfsrestore`(1M) command to recover a file system that has been corrupted, accidentally remade, or destroyed. For details about the syntax and options used in the procedure, see the `samfsdump`(1M) man page.

▼ To Restore a File System Using File System Manager

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

   The File Systems Summary page is displayed.

2. **Select the radio button next to the file system for which you want to restore files.**

3. **From the Operations drop-down menu, choose Restore.**

   The Restore File System page is displayed.

4. **If the metadata snapshot file is displayed as a link in the Metadata Snapshot Summary table, skip to the next step. Otherwise, make the snapshot available by selecting the radio button next to the snapshot that is unavailable and click Make Available for Browsing.**

5. **Perform either of the following in the Metadata Snapshot Summary table:**

   - Click a metadata snapshot file to browse its contents.
   - Select the radio button next to a metadata snapshot file and click Browse.

   The Restore File System page is refreshed and the top-level items in the selected metadata snapshot are displayed in the Metadata Snapshot Entries table.

6. **Under Restore Type select Entire File System.**

7. **From the Online Status After Restoring drop-down menu, choose how you want the file restored.**

8. **Click Restore.**

Note – The File System Manager software can only restore files from snapshots created by the File System Manager software.

▼ To Restore a File System Using the Command Line Interface

This example restores a file system from a `samfsdump` dump file called `/dump_sam1/dump/041126`.

1. **Use the `cd`(1M) command to change to the mount point for the file system or to the directory location where you want to restore the file system.**

Caution – Consider restoring the file system first into a temporary directory and verifying that the restoration succeeds before restoring directly into the existing file system. This removes the risk of destroying the current file system before you can be sure the restoration is going to work. If the restoration fails, the file system may be recoverable by some other process.

In the following example, the mount point is `/sam1`.

```
# cd /sam1
```

2. **Use the `samfsrestore` command with the `-T` and `-f` options to restore the entire file system relative to the current directory.**

   Use the syntax shown in the following screen example, specifying the pathname of the dump file after the `-f` option, and the pathname of the restore log file after the `-g` option.

```
# samfsrestore -T -f /dump_sam1/dumps/041126 -g log
```

Note – The *log* file that is created in the previous screen example can be used as input to `restore.sh`(1M) script to stage back files that were online at the time of the dump.

Recovering a SAM-QFS File System Without a Dump File

You may be able to recover data from a SAM-QFS file system even if you do not have access to output from a `samfsdump`(1M) command, or to an archiver log file.

The following procedure shows you how to recreate user files by reloading tape or optical disk and using the `star`(1M) command's `-n` option.

Note – Recovering file systems from archive cartridges and using the `star` command is a tedious and time-consuming process. This should not be considered the normal condition for disaster recovery.

▼ To Recover Without a Dump File

1. **(Optional) Disable any automated processes that are related to Sun StorEdge SAM-FS operations.**

If you have any of the following automated processes running, disable them during the recovery process to ensure that no data is lost:

- Recycling. Disable any recycling activities, including those triggered by an entry in root's `crontab`(4). Failure to disable recycling activity could result in tapes being recycled and relabeled that contain active data.
- Archiving
- Processes that capture `samfsdump`(1M) files. Suspending these processes saves an existing `samfsdump` output file, and provides an opportunity for easier recovery.
- Writes into the file system

2. **(Optional) Disable NFS-sharing for the file system.**

   It can be easier to recover data if the file system is not NFS-sharing the file systems during the recovery period.

3. **Use the `sammkfs`(1M) command to remake the SAM-QFS file system to be restored.**

4. **Identify the cartridges that contain the archive copy information.**

5. **Read all the archive media.**

   If you are using tapes, use `tar`(1M), `gnutar`(1M), or `star`(1M).

6. **If recovering from tape media, use the `tarback.sh` script.**

   The `tarback.sh`(1M) script is described in "Disaster Recovery Commands and Tools" on page 53. For more information about this script, see the tarback.sh man page. See also "Unreadable Label – No Other Copies Available" on page 112 for an example of how to use the script.

   The script is located in `/opt/SUNWsamfs/examples/tarback.sh`. This script identifies a single tape drive for use during recovery, and provides a list of VSNs to recover. The script uses `star`(1M) to loop through a volume, reading all available archive files.

   The `star`(1M) command is an enhanced version of `gnutar`(1M). The `tarback.sh` script uses `star`(1M) and the `-n` option, which is a `star`(1M) extension to `gnutar`(1M). The `-n` option restores only files that are newer than the existing copy. If the archive copy you are about to restore is older than the existing copy, the restore is skipped. This is important because it means that you do not have to worry about reading archive media in a specific order.

7. **If recovering from magnetic-optical media, contact Sun support.**

CHAPTER **8**

Recovering From Catastrophic Failure

Certain events can be classified as catastrophic failures. These include the damage caused by natural disasters, such as flooding in a computer room. This chapter provides a procedure to follow after such an event. You might require the assistance from your ASP or from Sun Microsystems customer support to successfully complete the procedures described in this chapter.

▼ To Recover From a Catastrophic Failure

Any system component, software element, or SAM-QFS file system that has not failed should not be recovered. However, you might need to reconfigure the SAM-QFS file system on a restored system to regain access to file systems or to determine whether any file system has failed. For details in performing these tasks, see the other chapters of this manual.

1. **Determine the failed system component.**

   See "To Restore Failed System Components" on page 118.

2. **Disable the archiver and the recycler until all files are restored.**

   See "To Disable the Archiver and Recycler Until All Files Are Restored" on page 118.

3. **Compare previous and current configuration files, and reconcile inconsistencies.**

   See "To Keep and Compare Previous and Current Configuration and Log Files" on page 120.

4. **Repair disks.**

   See "To Repair Disks" on page 121.

5. **Restore or build new library catalog files.**

   See "To Restore or Build New Library Catalog Files" on page 121.

6. **Make new file systems and restore from `samfsdump` output.**

   See “To Make New File Systems and Restore From samfsdump Output” on page 121.

▼ To Restore Failed System Components

1. **Ascertain which components have failed.**

   The following steps describe how to restore the following types of components:

   - Hardware
   - Operating environment
   - Sun StorEdge SAM-FS or Sun StorEdge QFS packages.

2. **If a hardware component has failed, restore it to operation, preserving any available data.**

   If the failing component is a disk drive that has not totally failed, preserve any information possible. Before replacing or reformatting the disk, identify any salvageable files (including those in the following list), and copy these files to a tape or to another disk for future use in the recovery process.

   - SAM-QFS file system dumps
   - Sun StorEdge SAM-FS configuration files, archiver log files, or library catalogs

3. **If the Solaris operating environment has failed, restore it to operation.**

   See “Recovering From Failure of the Operating Environment Disk” on page 22. Verify that the Solaris operating environment is functioning correctly before proceeding.

4. **If the Sun StorEdge SAM-FS or Sun StorEdge QFS packages have been damaged, remove and reinstall them from a backup copy or from its distribution file.**

   You can verify whether a package has been damaged by using the `pkgchk`(1M) utility.

5. **If disk hardware used by the Sun StorEdge SAM-FS software was repaired or replaced in Step 2, configure the disks (RAID binding or mirroring) if necessary.**

   Reformat disks only if they have been replaced or if it is otherwise absolutely necessary, because reformatting destroys all the file system information.

▼ To Disable the Archiver and Recycler Until All Files Are Restored

---

Caution – If the recycler is enabled so that it runs before all files are restored, cartridges with good archive copies may be improperly relabeled.

---

1. **Add a single global `wait` directive to the `archiver.cmd` file or add a file-system-specific wait directive for each file system for which you want to disable archiving.**

Note – The `wait` directive can be applied globally or individually to one or more file systems.

---

a. **Open the `/etc/opt/SUNWsamfs/archiver.cmd` file for editing and find the section where you want to insert the `wait` directive.**

   CODE EXAMPLE 8-1 shows using the `vi(1)` command to edit the file. In the example, local archiving directives exist for two file systems `samfs1` and `samfs2`.

CODE EXAMPLE 8-1 Example `archiver.cmd` File

```
# vi /etc/opt/SUNWsamfs/archiver.cmd
...
fs = samfs1
allfiles   .
1   10s
fs = samfs2
allfiles   .
1   10s
```

b. **Add the `wait` directive.**

   CODE EXAMPLE 8-2 shows a global `wait` directive inserted before the first `fs =` command (`fs = samfs1`).

CODE EXAMPLE 8-2 Example `archiver.cmd` File with a Global `wait` Directive

```
wait
fs = samfs1
allfiles   .
1   10s
fs = samfs2
allfiles   .
1   10s
:wq
```

CODE EXAMPLE 8-3 shows two file system-specific `wait` directives inserted after the first and second `fs =` commands (`fs = samfs1` and `fs = samfs2`).

CODE EXAMPLE 8-3 Example `archiver.cmd` File with File System-specific `wait` Directives

```
fs = samfs1
wait
allfiles   .
1   10s
fs = samfs2
```

CODE EXAMPLE 8-3 Example `archiver.cmd` File with File System-specific `wait` Directives *(Continued)*

```
wait
allfiles   .
1   10s
:wq
```

2. **Add a global `ignore` directive to the `recycler.cmd` file or add a file-system-specific `ignore` directive for each library for which you want to disable recycling.**

   a. **Open the `/etc/opt/SUNWsamfs/recycler.cmd` file for editing.**

   CODE EXAMPLE 8-4 shows using the `vi(1)` command to edit the file.

CODE EXAMPLE 8-4 Example `recycler.cmd` File

```
# vi /etc/opt/SUNWsamfs/recycler.cmd
...
         logfile = /var/adm/recycler.log
         lt20 -hwm 75 -mingain 60
         lt20 75 60
         hp30 -hwm 90 -mingain 60 -mail root
         gr47 -hwm 95 -mingain 60 -mail root
```

   b. **Add the `ignore` directives.**

   CODE EXAMPLE 8-5 shows `ignore` directives added for three libraries.

CODE EXAMPLE 8-5 Example `recycler.cmd` File with `ignore` Directives

```
#  recycler.cmd.after  - example recycler.cmd file
#
         logfile = /var/adm/recycler.log
         lt20 -hwm 75 -mingain 60 -ignore
         hp30 -hwm 90 -mingain 60 -ignore -mail root
         gr47 -hwm 95 -mingain 60 -ignore -mail root
```

▼ To Keep and Compare Previous and Current Configuration and Log Files

1. **Recover any available Sun StorEdge SAM-FS configuration files or archiver log files from the system's disks before rebuilding the system.**

2. **Compare the restored versions of all configuration files represented in the `SAMreport` with those restored from the system backups.**

3. **If inconsistencies exist, determine the effect of the inconsistencies and reinstall the Sun StorEdge QFS file system, if necessary, using the configuration information in the `SAMreport`.**

   For more information on `SAMreport` file, see the `samexplorer`(1M) man page.

▼ To Repair Disks

- **For SAM-QFS file systems that reside on disks that have not been replaced, run the `samfsck`(1M) utility to repair small inconsistencies, reclaim lost blocks, and so on.**

  For command line options to the `samfsck` utility, see the `samfsck`(1M) man page.

▼ To Restore or Build New Library Catalog Files

1. **Replace the most recent library catalog file copies from the removable media files, from the Sun StorEdge SAM-FS server disks, or from the most recent file system archive copies (which are likely to be slightly out of date).**

2. **If the library catalogs are unavailable, build new catalogs by using the `build.cat`(1M) command and the library catalog section of the most recent `SAMreport` as input.**

   Use the newest library catalog copy available for each automated library.

Note – Sun StorEdge SAM-FS systems automatically rebuild library catalogs for SCSI-attached automated libraries. This does not occur for ACSLS-attached automated libraries. Tape usage statistics are lost.

▼ To Make New File Systems and Restore From samfsdump Output

For those SAM-QFS file systems that were resident (partially or totally) on disks that were replaced or reformatted, perform the following procedure.

1. **Obtain the most recent copy of the `samfsdump`(1M) output file.**

2. **Make a new file system and restore the SAM-QFS file system using the `samfsdump` output file.**

a. **Use the `sammkfs`(1M) command to make a new file system.**

CODE EXAMPLE 8-6 shows this process.

CODE EXAMPLE 8-6 Using the `sammkfs`(1M) Command

```
# mkdir /sam1
# sammkfs samfs1
# mount samfs1
```

b. **Use the `samfsrestore`(1M) command with the `-f` option and the `-g` option.**

Specify the location of the `samfsdump` output file after the `-f` option. Specify the name of a log file after the `-g` option. The `-g` option creates a log of the files that had been online. The following example shows this:

```
# cd /sam1
# samfsrestore -f /dump_sam1/dumps/040120 -g /var/adm/messages/restore_log
```

Note – Once all file systems have been restored, the system can be made available to users in degraded mode.

---

3. **On the file systems restored in Step 2, perform the following steps:**

a. **Run the `restore.sh`(1M) script against the log file created in Step b of Step 2, and stage all files that were known to be online before the outage.**

b. **Run the `sfind`(1M) command against the SAM-QFS file system to determine which files are labeled as damaged.**

These files might or might not be restorable from tape, depending on the content of the archive log files. Determine the most recently available archive log files from one of the following sources:

- The removable media file.
- The Sun StorEdge SAM server disk.
- The most recent file system archive if not available from either of the previous two sources. This source is likely to be slightly outdated.

c. **Run the `grep`(1) command against the most recent archive log file to search for the damaged files, to determine whether any of the damaged files were archived to tape since the last time the `samfsdump`(1M) command was run.**

d. **Examine the archive log files to identify any archived files that do not exist in the file system.**

   e. **Use the `star`(1M) command to restore files from the archive media and to restore files that have been labeled as damaged.**

      These are files identified in Step c and Step d.

4. **Reimplement disaster recovery scripts, methods, and `cron`(1M) jobs using information from the backup copies.**

Glossary

A

addressable storage	The storage space encompassing online, nearline, offsite, and offline storage that is user-referenced through a Sun StorEdge SAM-FS or SAM-QFS file system.
archive media	The media to which an archive file is written. Archive media can be removable tape or magneto-optical cartridges in a library. In addition, archive media can be a mount point on another system.
archiver	The archive program that automatically controls the copying of files to removable cartridges.
archive storage	Copies of file data that have been created on archive media.
audit (full)	The process of loading cartridges to verify their VSNs. For magneto-optical cartridges, the capacity and space information is determined and entered into the automated library's catalog.
automated library	A robotically controlled device designed to automatically load and unload removable media cartridges without operator intervention. An automated library contains one or more drives and a transport mechanism that moves cartridges to and from the storage slots and the drives.

B

backup storage	A snapshot of a collection of files for the purpose of preventing inadvertent loss. A backup includes both the file's attributes and associated data.

block allocation map	A bitmap representing each available block of storage on a disk and indicating whether the block is in use or free.
block size	See *DAU*.

C

cartridge	A physical entity that contains media for recording data, such as a tape or optical disk. Sometimes referred to as *a piece of media*, *a volume*, or *the medium*.
catalog	A record of the VSNs in an automated library. There is one catalog for each automated library and, at a site, there is one historian for all automated libraries.
client-server	The model of interaction in a distributed system in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called the client. The program satisfying the response is called the server.
connection	The path between two protocol modules that provides reliable stream delivery service. A TCP connection extends from a TCP module on one machine to a TCP module on the other.

D

data device	In a file system, a device or group of devices upon which file data is stored.
DAU	Disk allocation unit. The basic unit of online storage. Also called *block size*.
device logging	A configurable feature that provides device-specific error information used to analyze device problems.
device scanner	Software that periodically monitors the presence of all manually mounted removable devices and that detects the presence of mounted cartridges that can be requested by a user or other process.
direct access	A file attribute (stage never) designating that a nearline file can be accessed directly from the archive media and need not be retrieved to disk cache.
direct-attached library	An automated library connected directly to a server using a SCSI interface. A SCSI-attached library is controlled directly by the SAM-QFS software.

direct I/O	An attribute used for large block-aligned sequential I/O. The `setfa`(1) command's `-D` option is the direct I/O option. It sets the direct I/O attribute for a file or directory. If applied to a directory, the direct I/O attribute is inherited.
directory	A file data structure that points to other files and directories within the file system.
disk allocation unit	See *DAU*.
disk buffer	In a Sun SAM-Remote configuration, the buffer on the server system that is used for archiving data from the client to the server.
disk cache	The disk-resident portion of the file system software, used to create and manage data files between online disk cache and archive media. Individual disk partitions or an entire disk can be used as disk cache.
disk space threshold	The maximum or minimum level of disk cache utilization, as defined by an administrator. The releaser controls disk cache utilization based on these predefined disk space thresholds.
disk striping	The process of recording a file across several disks, thereby improving access performance and increasing overall storage capacity. See also *striping*.
drive	A mechanism for transferring data to and from a removable media volume.

E

Ethernet	A local-area, packet-switched network technology. Originally designed for coaxial cable, it is now found running over shielded, twisted-pair cable. Ethernet is a 10- or 100-Mbytes/second LAN.
extent array	The array within a file's inode that defines the disk location of each data block assigned to the file.

F

family device set	See *family set*.
family set	A storage device that is represented by a group of independent physical devices, such as a collection of disks or the drives within an automated library. See also *storage family set*.

FDDI	Fiber-distributed data interface is a standard for data transmission in a local area network that can extend in range up to 200 km (124 miles). The FDDI protocol is based on the token ring protocol.
Fibre Channel	The ANSI standard that specifies high-speed serial communication between devices. Fibre Channel is used as one of the bus architectures in SCSI-3.
file system	A hierarchical collection of files and directories.
file-system-specific directives	Archiver and releaser directives that follow global directives in the `archiver.cmd` file, are specific to a particular file system, and begin with `fs =`. File-system-specific directives apply until the next `fs =` directive line or the end of file is encountered. If multiple directives affect a file system, the file-system-specific directives override the global directives.
FTP	File transfer protocol. An Internet protocol for transferring files between two hosts over a TCP/IP network.

G

global directives	Archiver and releaser directives that apply to all file systems and that appear before the first `fs =` line.
grace period	For disk quotas, the amount of time for which a user is allowed to create files and allocate storage after reaching the soft limit.

H

hard limit	For disk quotas, the maximum limit on file system resources, blocks, and inodes that users cannot exceed.

I

indirect block	A disk block that contains a list of storage blocks. File systems have up to three levels of indirect blocks. A first-level indirect block contains a list of blocks used for data storage. A second-level indirect block contains a list of first-level indirect blocks. A third-level indirect block contains a list of second-level indirect blocks.

inode	Index node. A data structure used by the file system to describe a file. An inode describes all the attributes associated with a file other than the name. The attributes include ownership, access, permission, size, and the file location on the disk system.
inode file	A special file (`.inodes`) on the file system that contains the inode structures for all files resident in the file system. Inodes are 512 bytes long. The inode file is a metadata file, which is separated from file data in the file system.

K

kernel	The central controlling program that provides basic system facilities. The UNIX kernel creates and manages processes, provides functions to access the file system, provides general security, and supplies communication facilities.

L

LAN	Local area network.
lease	A function that grants a client host permission to perform an operation on a file for a specified period of time. The metadata server issues leases to each client host. The leases are renewed as necessary to permit continued file operations.
library	See *automated library*.
library catalog	See *catalog*.
local file system	A file system that is installed on one node of a Sun Cluster system and is not made highly available to another node. Also, a file system that is installed on a standalone server.
LUN	Logical unit number.

M

`mcf`	Master configuration file. The file that is read at initialization time that defines the relationships between the devices (the topology) in a file system environment.

media	Tape or optical disk cartridges.
media recycling	The process of recycling or reusing archive media with few active files.
metadata	Data about data. Metadata is the index information used to locate the exact data position of a file on a disk. It consists of information about files, directories, access control lists, symbolic links, removable media, segmented files, and the indexes of segmented files.
metadata device	A device (for example, a solid-state disk or mirrored device) upon which file system metadata is stored. Having file data and metadata on separate devices can increase performance. In the `mcf`(4) file, a metadata device is declared as an `mm` device within an `ma` file system.
mirror writing	The process of maintaining two copies of a file on disjointed sets of disks to prevent loss from a single disk failure.
mount point	The directory on which a file system is mounted.
multireader file system	A single-writer, multireader capability that enables you to specify a file system that can be mounted on multiple hosts. Multiple hosts can read the file system, but only one host can write to the file system. Multiple readers are specified with the `-o reader` option with the `mount`(1M) command. The single-writer host is specified with the `-o writer` option with the `mount`(1M) command. For more information on the `mount`(1M) command, see the `mount_samfs`(1M) man page.

N

name space	The metadata portion of a collection of files that identifies the file, its attributes, and its storage locations.
nearline storage	Removable media storage that requires robotic mounting before it can be accessed. Nearline storage is usually less expensive than online storage, but it takes somewhat longer to access.
network-attached automated library	A library, such as those from StorageTek, ADIC/Grau, IBM, or Sony, that is controlled using a software package supplied by the vendor. The SAM-QFS file system interfaces with the vendor software using a SAM-QFS 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 SAM-QFS 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 SAM-QFS 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 SAM-QFS software with the Sun StorEdge SAM-FS 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 SAM-QFS command set as well as standard UNIX file system commands.
`samfsrestore`	A program that restores inode and directory information from a control structure dump. See also *samfsdump*.
SCSI	Small Computer System Interface. An electrical communication specification commonly used for peripheral devices such as disk and tape drives and automated libraries.

small computer system interface	See SCSI.
soft limit	For disk quotas, a threshold limit on file system resources (blocks and inodes) that you can temporarily exceed. Exceeding the soft limit starts a timer. When you exceed the soft limit for the specified time, no further system resources can be allocated until you reduce file system use to a level below the soft limit.
staging	The process of copying a nearline or offline file from archive storage back to online storage.
storage family set	A set of disks that are collectively represented by a single disk family device.
storage slots	Locations inside an automated library in which cartridges are stored when not being used in a drive. If the library is direct-attached, the contents of the storage slots are kept in the automated library's catalog.
striped group	A collection of devices within a file system that is defined in the `mcf`(4) file as one or more `g`*XXX* devices. Striped groups are treated as one logical device and are always striped with a size equal to the disk allocation unit (DAU).
stripe size	The number of disk allocation units (DAUs) to be allocated before writing proceeds to the next device of a stripe. If the `stripe=0` mount option is used, the file system uses round-robin access, not striped access.
striping	A data access method in which files are simultaneously written to logical disks in an interlaced fashion. SAM-QFS file systems provide two types of striping: "hard striping," using stripe groups, and "soft striping," using the `stripe=`*x* mount parameter. Hard striping is enabled when a file system is set up, and requires the definition of stripe groups within the `mcf`(4) file. Soft striping is enabled through the `stripe=`*x* mount parameter, and can be changed for the file system or for individual files. It is disabled by setting `stripe=0`. Hard and soft striping can both be used if a file system is composed of multiple stripe groups with the same number of elements. See also *round robin*.
Sun SAM-Remote client	A SAM-QFS 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 SAM-QFS storage management server and a Sun SAM-Remote server daemon that defines libraries to be shared among Sun SAM-Remote clients.
superblock	A data structure in the file system that defines the basic parameters of the file system. The superblock is written to all partitions in the storage family set and identifies the partition's membership in the set.

T

tar	Tape archive. A standard file and data recording format used for archive images.
TCP/IP	Transmission Control Protocol/Internet Protocol. The internet protocols responsible for host-to-host addressing and routing, packet delivery (IP), and reliable delivery of data between application points (TCP).
timer	Quota software that keeps track of the period starting when a user reaches a soft limit and ending when the hard limit is imposed on the user.

V

volume	A named area on a cartridge for sharing data. A cartridge has one or more volumes. Double-sided cartridges have two volumes, one on each side.
volume overflow	A capability that enables the system to span a single file over multiple volumes. Volume overflow is useful for sites using very large files that exceed the capacity of their individual cartridges.
VSN	Volume serial name. In the context of archiving to removable media cartridges, the VSN is a logical identifier for magnetic tape and optical disk that is written in the volume label. In the context of archiving to disk cache, this is the unique name for the disk archive set.

W

WORM	Write once read many. A storage classification for media that can be written only once but read many times.

Index