PART NUMBER
312520101

VERSION NUMBER
4.0

EDITION NUMBER
1

ASM

Application Storage Manager ™

PRODUCT TYPE
SOFTWARE

STORAGE AND ARCHIVE MANAGER GUIDE (UNIX)

For ASM, ASM-QFS, and ASM/QFS-Standalone

StorageTek®

Application Storage Manager™ (ASM)

ASM, ASM-QFS, and ASM/QFS-Standalone Storage and Archive Manager Guide

Version 4.0

for Unix

First Edition

PN 312520101

Information contained in this publication is subject to change without notice. Comments concerning the contents of this publication should be directed to:

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

First Edition (May 2003)

Contents

Preface

This manual, the *ASM, ASM-QFS, and ASM/QFS-Standalone Storage and Archive Management Guide*, describes the storage and archive management software supported within the StorageTek ASM and ASM-QFS 4.0 releases. This software automatically copies files from online disk to archive media. The archive media can consist of either online disk or removable media cartridges.

The ASM and ASM-QFS 4.0 releases are supported in the Sun Solaris™ operating environment (OE) 7, 8, and 9 releases.

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

How This Book Is Organized

This manual contains the following chapters:

- Chapter 1 provides overview information.
- Chapter 1 explains basic operations. The information in this chapter applies to most automated libraries and manually loaded devices.
- Chapter 2 explains how to manage cartridges in libraries with operational instructions that pertain only to that kind of library. This chapter describes these libraries and their library-specific basic operational procedures.
- Chapter 4 explains the archiving process.
- Chapter 5 explains the releasing process.
- Chapter 6 explains the staging process.
- Chapter 7 explains the recycling process.
- Chapter 8 describes how to use the graphical user interface (GUI) tools that can be used with the ASM and ASM-QFS software.
- Chapter 9 describes how to use the `samu`(1M) operator utility.
- Chapter 10 describes upgrade procedures that are specific to ASM and ASM-QFS environments.
- Chapter 11 describes advanced topics in ASM and ASM-QFS operations.
- The glossary defines terms used in this and other ASM/QFS-Standalone, ASM, and ASM-QFS documentation.

Related Documentation

This manual is part of a set of documents that describes the operations of the ASM/QFS-Standalone, ASM, and ASM-QFS software products. The following table shows the complete release 4.0 documentation set for these products.

Table 1.

Title	Part Number
ASM, ASM-QFS, and ASM/QFS-Standalone Storage and Archive Management Guide	312520101
ASM-Remote Administrator's Guide	312520201
ASM, ASM-QFS, and ASM/QFS-Standalone Installation and Configuration Guide	312502301
ASM, ASM-QFS, and ASM/QFS-Standalone File System Administrator's Guide	312502401
ASM, ASM-QFS, and ASM/QFS-Standalone Disaster Recovery Guide	312502501

Note that the *ASM-Remote Administrator's Guide* has not been updated for the 4.0 release. An updated version of this manual will be provided at a later date.

How to Obtain Documentation

All the ASM publications are available from the following

sources:

- Contact StorageTek Publication Sales and Service at 800-436-5554 or send a fax to 303-661-7367.
- Online (for viewing and printing), at the StorageTek Customer Resource Center (CRC) website at: www.support.storagetek.com. Click on Software and go to the ASM Software list.

Access to the CRC site requires a password. To obtain a password, call StorageTek Customer Support at 1-800-678-4430.

Licensing

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

Diagnostics

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

Support

The publication “Requesting Software Support” is included in your media package. Please consult this book for the most information on your ASM support options, as well as regional phone numbers and procedures.

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.

See one or more of the following for this information:

- *Solaris Handbook for Sun Peripherals*
- AnswerBook2™ online documentation for the Sun Solaris OE
- Other software documentation that you received with your system

Typographic Conventions

Table 2. lists the typographic conventions used in this manual.

Table 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:`

Table 2. Typographic Conventions

Typeface or Symbol	Meaning	Examples
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`

Shell Prompts

Table 3. shows the shell prompts that this manual uses.

Table 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	#

Overview

1

The ASM and ASM-QFS environments provide a configurable file system with storage, archive management, and retrieval capabilities. The ASM and ASM-QFS software archives files by copying the files from online disk cache to archive media. The archive media can consist of disk slices in another file system or can consist of removable tape or magneto-optical cartridges in automated or manually loaded storage devices. In addition, the ASM and ASM-QFS software products automatically maintain online disk space at site-specified usage thresholds. These products release disk space associated with archived file data and restore the files to online disk when they are needed.

This chapter provides a technical overview of the ASM and ASM-QFS components. The following topics are presented:

Capabilities

The ASM environment includes a file system and the storage and archive management software. The ASM-QFS environment includes the QFS file system. All file systems are high-performance UNIX file systems that reside in the server's disk cache. The major difference between these file systems is that the ASM-QFS file system offers more high-performance features, including a distributed shared file system. For more information about the file systems themselves, see the *ASM, ASM-QFS, and ASM/QFS-Standalone File System Administrator's Guide*.

The other components that reside in the ASM and ASM-QFS environments are as follows:

- The archiver, which automatically copies online disk cache files to archive media. The archive media can consist of either online disk files or removable media cartridges.
- The releaser, which automatically maintains the file system's online disk cache at site-specified percentage usage thresholds by freeing disk blocks occupied by eligible archived files.

- The stager, which restores file data to the disk cache. When a user or process requests file data that has been released from disk cache, the stager automatically copies the file data back to the online disk cache.
- The recycler, which clears archive volumes of expired archive copies and makes volumes available for reuse.

The following sections briefly describe each of these capabilities. More information on these capabilities can be found in subsequent chapters.

Archiving

By default, the archiver automatically creates one archive copy of all files in a ASM or ASM-QFS file system and copies the files to archive media. You can configure the archiver to create up to four archive copies on a variety of archive media. If a file is *segmented*, each segment is treated as a file, and each segment is archived separately. The archiving process is initiated after disk-based files match a site-definable set of selection criteria.

For more information on the archiver, see “Archiving” on page 77. For more information on segmented files, see “Advanced Topics” on page 269.

Releasing

Releasing is the process of freeing primary (disk) storage that is used by an archived file’s data. Two threshold values, both expressed as a percentage of total disk space, are used to manage online disk cache free space. These thresholds are the high water mark and the low water mark. When online disk consumption exceeds the high water mark, the system automatically begins releasing the disk space occupied by eligible archived files. Disk space occupied by archived file data is released until the low water mark is reached. Files are selected for release depending on the file’s size and age. Optionally, the first portion of a file can be retained on disk for speedy access and for masking staging delays. If a file has been archived in segments, portions of the file can be released individually. For more information on the releaser, see “Releasing” on page 145.

Staging

When a file whose data blocks have been released is accessed, the stager automatically stages the file or file segment data back to online disk cache. The read operation tracks along directly behind the staging operation, allowing the file to be immediately available to an application before the entire file is completely staged.

The ASM and ASM-QFS software processes stage request errors automatically. If a stage error is returned, the system attempts to find the next available archive copy of the file, if there is another copy and if there is a device available to read the copy’s media. Stage errors that can be automatically processed include media errors, unavailability of media, unavailability of an automated library, and others. For more information on staging, see “Staging” on page 163.

Recycling

As users modify files, archive copies associated with the old versions of these files are considered to be *expired* on their archive media and can be purged from the system. The recycler identifies the archive volumes with the largest proportions of expired archive copies and directs the movement of nonexpired copies to separate volumes.

If only expired copies exist on a given volume, a site-defined action is performed. For example, a removable media volume can be relabeled for immediate reuse, or it can be exported to offsite storage as an historical record of file changes. System administrators can restore previous versions of files from expired archive copies using standard UNIX utilities. The recycling process is transparent to end users as it relates to their data files. For more information on recycling, see “Recycling” on page 177.

Storage Devices

The ASM and ASM-QFS environments support a wide variety of tape storage and magneto-optical devices. For a list of supported storage devices, contact your StorageTek sales representative or your authorized service provider (ASP).

The relationships between the devices managed within the ASM or ASM-QFS environment are defined in the master configuration file, `/etc/opt/SUNWsamfs/mcf`. The `mcf` file specifies the removable media devices, libraries, and file systems included in the ASM and ASM-QFS environment. Each piece of equipment is assigned a unique equipment identifier in the `mcf` file. Entries in the `mcf` also define manually mounted archiving devices and automated library catalog files.

When possible, the system uses the standard Sun Solaris disk and tape device drivers. For devices not directly supported in the Sun Solaris OE, such as certain library and optical disk devices, special device drivers are included in the ASM and ASM-QFS software packages.

Commands

The ASM and ASM-QFS environments consist of a file system, daemons, processes, various types of commands (user, administrator, and so on), and tools. Overview information for the daemons is described in the *ASM, ASM-QFS, and ASM/QFS-Standalone File System Administrator’s Guide*, but individual daemons are described throughout the documentation set where appropriate. This section introduces the commands, but see the man pages that are included in the software distribution for more information.

This section contains the following topics:

User Commands

By default, file system operations are transparent to the end user. Depending on your site practices, however, you might want to make some commands available to users at your site to fine-tune certain operations. Table 1. summarizes these commands.

Table 1. User Commands

Command	Description	Used By
`archive`(1)	Archives files and sets archive attributes on files.	ASM, ASM-QFS
`release`(1)	Releases disk space and sets release attributes on files.	ASM, ASM-QFS
`request`(1)	Creates a removable media file.	ASM, ASM-QFS
`sdu`(1)	Summarizes disk usage. The `sdu`(1) command is based on the GNU version of the `du`(1) command.	ASM/QFS-Standalone, ASM, ASM-QFS
`segment`(1)	Sets segmented file attributes.	ASM, ASM-QFS
`setfa`(1)	Sets file attributes.	ASM/QFS-Standalone, ASM, ASM-QFS

Table 1. User Commands (Continued)

Command	Description	Used By
`sfind`(1)	Searches for files in a directory hierarchy. The `sfind`(1) command is based on the GNU version of the `find`(1) command and contains options for displaying file system options.	ASM/QFS-Standalone, ASM, ASM-QFS
`sls`(1)	Lists contents of directories. The `sls`(1) command is based on the GNU version of the `ls`(1) command and contains options for displaying file system attributes and information.	ASM/QFS-Standalone, ASM, ASM-QFS
`squota`(1)	Reports quota information.	ASM/QFS-Standalone, ASM, ASM-QFS
`ssum`(1)	Sets the checksum attributes on files.	ASM, ASM-QFS
`stage`(1)	Sets stage attributes on files and copies offline files to disk.	ASM, ASM-QFS

General System Administrator Commands

Table 2. summarizes the commands that provide system management and maintenance capabilities.

Table 2. General System Administrator Commands

Command	Description	Used By
`samcmd`(1M)	Executes one `samu`(1M) operator interface utility command.	ASM/QFS-Standalone, ASM, ASM-QFS
`samd`(1M)	Starts or stops robotic and removable media daemons.	ASM, ASM-QFS
`samset`(1M)	Changes ASM or ASM-QFS settings.	ASM, ASM-QFS
`samu`(1M)	Invokes the full-screen, text-based operator interface. This interface is based on the `curses`(3X) software library. The `samu` utility displays the status of devices and allows the operator to control automated libraries.	ASM/QFS-Standalone, ASM, ASM-QFS

File System Commands

Table 3. summarizes the file system commands. These are used to perform file system maintenance operations.

Table 3. File System Commands

Commands	Description	Used By
`mount`(1M)	Mounts a file system. The man page name for this command is `mount_samfs`(1M).	ASM/QFS-Standalone, ASM, ASM-QFS
`qfsdump`(1M) `qfsrestore`(1M)	Creates or restores a dump file containing the file data and metadata associated with an ASM/QFS file system.	ASM/QFS-Standalone
`sambcheck`(1M)	Lists block usage for a file system.	QFS-Standalone, ASM, ASM-QFS
`samchaid`(1M)	Changes file admin set ID attribute. For use with quotas.	ASM/QFS-Standalone, ASM, ASM-QFS
`samfsck`(1M)	Checks and repairs metadata inconsistencies in a file system and reclaims allocated, but unused, disk space.	ASM/QFS-Standalone, ASM, ASM-QFS
`samfsconfig`(1M)	Displays configuration information.	ASM/QFS-Standalone, ASM, ASM-QFS
`samfsdump`(1M) `samfsrestore`(1M)	Creates or restores a dump file of the metadata associated with a ASM or ASM-QFS file system.	ASM, ASM-QFS
`samfsinfo`(1M)	Displays information about the layout of an ASM/QFS-Standalone, ASM, or ASM-QFS file system.	ASM/QFS-Standalone, ASM, ASM-QFS
`samgrowfs`(1M)	Expands a file system by adding disk devices.	ASM/QFS-Standalone, ASM, ASM-QFS

Table 3. File System Commands (Continued)

Commands	Description	Used By
`sammkfs`(1M)	Initializes a new file system from disk devices.	ASM/QFS-Standalone, ASM, ASM-QFS
`samncheck`(1M)	Returns a full directory path name given the mount point and inode number.	ASM/QFS-Standalone, ASM, ASM-QFS
`samquota`(1M)	Reports, sets, or resets quota information.	ASM/QFS-Standalone, ASM, ASM-QFS
`samquotastat`(1M)	Reports on active and inactive file system quotas.	ASM/QFS-Standalone, ASM, ASM-QFS
`samsharefs`(1M)	Manipulates the QFS shared file system configuration information.	ASM/QFS-Standalone, ASM-QFS
`samtrace`(1M)	Dumps the trace buffer.	ASM/QFS-Standalone, ASM, ASM-QFS
`samunhold`(1M)	Releases SANergy file holds.	ASM/QFS-Standalone, ASM, ASM-QFS
`trace_rotate.sh`(1M)	Rotates trace files.	ASM/QFS-Standalone, ASM, ASM-QFS

Automated Library Commands

Table 4. summarizes the automated library commands that are used to configure, initialize, and maintain the automated libraries and devices within the ASM and ASM-QFS environments.

Table 4. Automated Library Commands

Command	Description
`auditslot`(1M)	Audits a single media cartridge slot within a specified automated library.
`build_cat`(1M)	Builds a media catalog file for an automated library. Optionally, can also populate the catalog file.
`chmed`(1M)	Sets or clears library catalog flags and values on a specific cartridge.
`cleandrive`(1M)	Requests that a tape drive be loaded with a cleaning tape.
`dump_cat`(1M)	Displays the content of a binary catalog file in various ASCII formats.
`import`(1M) `samexport`(1M)	Imports or exports cartridges from a library by placing it in the mailbox. For network-attached libraries, this command updates the library catalog, but it does not physically move cartridges.
`move`(1M)	Moves a cartridge from one slot to another.
`odlabel`(1M)	Labels optical disks for use with the ASM and ASM-QFS systems.
`samdev`(1M)	Adds `/dev/samst` logical device entries. Used to communicate automated library, optical disk, and tape drive information.
`tplabel`(1M)	Labels tapes for use with the ASM and ASM-QFS systems.

Archiver Commands

Table 5. summarizes the commands that control the archiver's actions within the ASM and ASM-QFS environments.

Table 5. Archiver Commands

Command	Description
`archiver`(1M)	Evaluates the archiver commands file for syntax completeness and semantic accuracy.

Table 5. Archiver Commands (Continued)

Command	Description
showqueue(1M)	Displays the content of an archiver queue file.
reserve(1M) unreserve(1M)	Reserves and unreserves volumes.

Specialized Maintenance Commands

Table 6. summarizes the various maintenance commands.

Table 6. Specialized Maintenance Commands

Command	Description
archive_audit(1M)	Generates a report of all archived files on each cartridge.
dev_down.sh(4)	Sends email to root when a device is marked down or off.
dmpshm(1M)	Dumps the shared memory segments.
exarchive(1M)	Manipulates (exchanges) archive copies.
itemize(1M)	Catalogs an optical disk.
load_notify.sh(1M)	Notifies the operator when the ASM or ASM-QFS software requests a cartridge that resides outside the library.
samload(1M) unload(1M)	Load or unloads a cartridge for a specified device.
rearch(1M) unrearch(1M)	Marks or unmarks archive entries to be rearchived.
sam-recycler(1M)	Reclaims space used by expired archive copies from archive media.
sam-releaser(1M)	Releases disk space from online disk cache file systems.
samdev(1M)	Creates symbolic links in the /dev/samst directory that point to the actual devices to be used by the ASM or ASM-QFS file system. This command is similar in function to the UNIX makedev(1M) command.
samset(1M)	Changes or displays variables used in ASM or ASM-QFS operations.

Table 6. Specialized Maintenance Commands (Continued)

Command	Description
set_admin.sh(1M)	Adds or removes permission for an administrator group to execute administrator commands.
set_state(1M)	Sets the state of a ASM or ASM-QFS device.
star(1M)	Creates tape archives and adds or extracts files. This is a GNU version of the tar(1) command, and it has been extended for use with the ASM or ASM-QFS file system. This command can be used for disaster recovery purposes when you need to read data from archive tapes.
unarchive(1M)	Deletes archive entries for one or more files.
undamage(1M)	Marks an archive entry for one or more files or directories as undamaged.

Application Programmer Interface

The application programmer interface (API) allows file system requests to be made from within a user application. The requests can be made locally or remotely to the machine on which the file system is running. The API consists of the libsam and libsamrpc libraries. These libraries contain library routines for obtaining file status, for setting archive, release, and stage attributes for a file, and for manipulating the library catalog of an automated library. The sam-rpcd server process handles remote requests. To automatically start the sam-rpcd server process, samrpc=on must be set in the defaults.conf file.

For more information on the API, see the intro_libsam(3) man page. This man page provides overview information for using the library routines in libsam and libsamrpc.

Operational Utilities

Within the ASM and ASM-QFS environments, the samu(1M) operator utility and the GUI tools can help you perform basic operations. Table 7. summarizes the GUI tools.

Table 7. Operational Tools

GUI Tools	Description
devicetool(1M)	Displays the status of devices and enables you to change their states.
libmgr(1M)	Displays information about and assists in managing automated libraries, devices, and mount requests within the ASM or ASM-QFS environment. This tool is based on Java technology.
previewtool(1M)	Displays and manages pending cartridge mount requests.
robottool(1M)	Displays the status of automated libraries; mounts and dismounts file systems; imports and exports cartridges; and displays library catalogs.
samtool(1M)	An application launcher that provides the starting point for accessing devicetool(1M), robottool(1M), and previewtool(1M).
samu(1M)	Provides the starting point for accessing the samu(1M) operator utility.

Basic Operations

2

An *automated library* is a robotically controlled device designed to load and unload removable cartridges without operator intervention. Cartridges are imported to and exported from the library. They are loaded and unloaded automatically. The archiving and staging processes use a site-defined scheme for allocating the number of drives to use. Automated libraries are also known as media changers, jukeboxes, robots, libraries, or media libraries.

The following sections describe aspects of using libraries in a ASM and ASM-QFS environment. The *ASM-QFS and ASM/QFS-Standalone Installation and Configuration Guide* provides initial configuration instructions, and this chapter provides operational instructions for automated libraries and manually loaded devices. In addition, this chapter describes the operator-oriented load notification facility that alerts an operator when a requested volume is not in a library.

Note: The ASM and ASM-QFS software interoperates with automated libraries from many manufacturers. Consult the README file distributed with this release for information pertinent to library model numbers, firmware levels, and other compatability information.

Some automated libraries have features that cause certain operations to differ from the ones described in this chapter. To determine whether your automated library has additional, vendor-specific operating instructions when used in a ASM or ASM-QFS environment, check “Vendor-Specific Operational Procedures for Libraries” on page 45.

This chapter contains the following topics:

■ Automated Library Hardware and Software

Libraries are attached to a ASM or ASM-QFS host system in one of the following ways:

- A direct attachment. A direct-attached library is connected directly to the host system using a small computer system interface (SCSI). This can be either a direct connection or a Fibre Channel connection. For example, a direct attachment is used for StorEdge libraries. The ASM- and ASM-QFS systems control these libraries directly using the SCSI standard for automated libraries.
- A network attachment. The ASM or ASM-QFS software can be configured as a client of the library's host system. The network-attached libraries include some of the StorageTek, ADIC/Grau, IBM, and Sony libraries. These libraries use a software package supplied by the vendor. In these cases, the ASM or ASM-QFS software interfaces with the vendor software using a daemon specifically designed for the automated library.

Table 8. shows the daemons specific to various automated libraries.

Table 8. Automated Library Daemons

Daemon	Description
`sam-robotsd`	Monitors the execution of robot control daemons. The `sam-robotsd` daemon is started automatically by the `sam-initd` daemon.
`sam-genericd`	Controls direct-attached libraries and media changers.
`sam-stkd`	Controls the StorageTek media changers through the ACSAPI interface.
`sam-ibm3494d`	Controls the IBM 3494 tape libraries through the `lmcpd` interface.
`sam-sonyd`	Controls the Sony network-attached automated libraries through the DZC-8000S interface.

Configuring Automated Libraries

You can configure automated libraries by editing the master configuration file found in `/etc/opt/SUNWsamfs/mcf`. The `Additional Parameters` field in the `mcf` file can contain the path name of a *library catalog file*, which contains information about the contents of each of the cartridges in the automated library's storage slots. By default, the library catalog file is written to `/var/opt/SUNWsamfs/`*family_set_name*.

Some automated libraries, such as the IBM 3494, the StorageTek network-attached libraries, the ADIC/Grau libraries, and the Sony network-attached libraries, require additional configuration information. This information is placed in a separate file, and the name of the file is also specified in the mcf file.

You should not configure ASM or ASM-QFS software for use with automated libraries until all vendor-supplied software (if any) is installed and known to be working. See the documentation supplied by the vendor, and be sure to you familiarize yourself with the vendor's software.

Parameters Files

If the automated library is using the vendor's software package running an interface as previously described, you must set up an additional parameters file that defines the system characteristics of the library and its drives. The parameters file differs for each vendor's automated library, so see the stk(7), ibm3494(7), ibm3584(7), grauaci(7), fujitsulmf(7), and sony(7) man pages for information on the vendor-specific parameters file.

■ The mcf File

The mcf file is formatted as described in the *ASM-QFS and ASM/QFS-Standalone Installation and Configuration Guide* and in the *ASM-QFS and ASM/QFS-Standalone File System Administrator's Guide*. Drives are associated with an automated library using a family set name. Each device is given a unique equipment ordinal, and so on.

The Equipment Identifier field differs depending on whether your library is direct-attacted or network-attached, as follows:

- For direct-attached libraries, the Equipment Identifier field is the /dev/samst entry for the automated library itself. See the configuration example for a magneto-optical automated library in the *ASM-QFS and ASM/QFS-Standalone Installation and Configuration Guide*.
- For network-attached automated libraries, such as network-attached Sony, StorageTek, ADIC/Grau, or Fujitsu automated libraries, the Equipment Identifier field must contain the full path name to the parameters file. For more information on the parameters file, see "Parameters Files" on page 15.

For both direct-attached and network-attached libraries, the Additional Parameters field can contain the full path name to the library catalog.

The `mcf` file shown in Figure 1. defines a StorageTek direct-attached automated library. In this example, two drives are associated with the automated library.

Figure 1. `mcf` **File Example**

```
# Equipment       Eq  Eq Family Dev Additional
# Identifier      Ord Ty Set    St  Parameters

# Define a file system

samfs1            10  ms samfs1
/dev/dsk/c1t1d0s0 11  md samfs1 on
/dev/dsk/c2t1d0s0 12  md samfs1 on

# Define a library

/dev/samst/c0t3u0 50  s9 stk    on
/dev/rmt/3cbn     51  sg stk    on
/dev/rmt/0cbn     52  sg stk    on
```

Periodically, you might want to change your `mcf` file to add new libraries, change file system disks, or perform other tasks. If changes are made to an `mcf` file, you must enter commands to reinitialize the `mcf` file and make the changes available to the ASM or ASM-QFS software. The procedures for reinitializing the `mcf` file are described in the *ASM-QFS and ASM/QFS-Standalone File System Administrator's Guide*.

The defaults.conf File

There are several directives you can set in the `/etc/opt/SUNWsamfs/defaults.conf` file to control automated library operations. The following sections describe some of these directives. For more information on these directives, see the `defaults.conf`(4) man page.

If you change the information in the `defaults.conf` file, for example, to accommodate changes in your site's library information, you must issue commands to reinitialize the file. The procedures for reinitializing the `defaults.conf` file are described in the *ASM-QFS and ASM/QFS-Standalone File System Administrator's Guide*.

The following lines from an example `defaults.conf` file show several parameters that can affect the configuration of an automated library:

```
exported_media = unavailable
attended = yes
tape = lt
log = LOG_LOCAL7
timeout = 300
# trace
# all on
# endtrace
labels = barcodes_low
lt_delay = 10
lt_unload = 7
lt_blksize = 256
```

Another sample file is located in `/opt/SUNWsamfs/examples/defaults.conf`. For more information on the `defaults.conf` file, see the `defaults.conf`(4) man page.

The `defaults.conf`(4) file allows you to configure many features in your environment. The following sections describe some of these features.

Barcodes

If you have a tape library that uses a barcode reader, you can configure the system to set the tape label equal to the first or last characters of the barcode label. You can accomplish this by setting the `labels` directive in the `defaults.conf` file, as shown in Table 9..

Table 9. The `labels` Directive in the `defaults.conf` File

Directive	Action
`labels = barcodes`	Uses the first six characters of the barcode as the label. This setting enables the archiver to label new media automatically if the tape is chosen. Default.
`labels = barcodes_low`	Uses the last six characters of the barcode as the label.
`labels = read`	Reads the label from the tape. This setting prevents the archiver from labeling new media automatically.

If `labels = barcodes` or `labels = barcodes_low` is in effect, the ASM or ASM-QFS system writes a label before the write is started for any tape that is mounted for a write operation that is write enabled, unlabeled, and has a readable barcode.

Drive Timing Values

You can set the load, unload, and unload wait time for devices using the *dev*_`delay` and *dev*_`unload` directives, respectively. These directives allow you to set these times to an interval that meets your site's requirements. These directives are set in the `defaults.conf` file.

The format of the *dev*_`delay` directive is as follows:

```
dev_delay = seconds
```

where:

dev	The device type as specified in the `mcf`(4) man page.
seconds	An integer number specifying the minimum elapsed time between a cartridge being loaded and the same cartridge's ability to be unloaded. The default is `30`.

The format of the *dev*_`unload` parameter is as follows:

```
dev_unload = seconds
```

where:

dev	The device type as specified in the `mcf`(4) man page.
seconds	An integer number specifying the amount of time to wait after an `unload` command is issued. This gives the automated library time to eject the cartridge, open the door, and perform other operations before the cartridge is removed. The default is `0`.

For example:

```
hp_delay = 10
lt_unload = 7
```

For more information on these directives, see the `defaults.conf`(4) man page.

Shared Drives

Typically, the ASM or ASM-QFS processes have exclusive control over a library's drives as declared in the host system's `mcf` file. In many cases, drives are defined in individual mcf files that are used by independant copies of ASM and ASM-QFS processes. If a process is not using a drive, the drive stays idle.

The shared drives capability allows two or more mcf files to define the same drive, and this makes the drive available to more than one ASM or ASM-QFS process. The shared drives capability allows each ASM or ASM-QFS process to share a drive, but it does not allow the processes to share media. Each ASM or ASM-QFS process must still maintain its own set of VSNs. A single piece of media cannot be shared among processes.

This feature can be useful, for example, if a library is attached to more than one host system in a ASM or ASM-QFS environment. The shared drives capability enables you to keep the drives in a library busy. The ASM or ASM-QFS processes coordinate the use of the drive.

Some libraries can be configured to share one or all media drives between multiple ASM or ASM-QFS processes on multiple host systems. To use shared drives, the library must be network attached. The following types of libraries support shared drives:

- StorageTek libraries that use the ACSLS interface
- IBM 3494 libraries that use the lmcpd interface
- Sony libraries that use the DZC-8000S interface

Each of the preceding libraries requires you to create a parameters file. To implement one or more shared drives, make sure that the shared keyword is specified in the parameters file for each drive that is to be shared. The placement of the shared keyword is specific to each manufacturer's library, but the following example parameters file shows how to specify the shared keyword for a StorageTek network-attached library:

```
#
# This is file: /etc/opt/SUNWsamfs/stk50
#
hostname = acsls_server_name
portnum = 50004
access = some_user  # No white space allowed in the user_id
field
/dev/rmt/0cbn = (acs=0, lsm=1, panel=0, drive=1) #a comment
/dev/rmt/1cbn = (acs=0, lsm=1, panel=0, drive=2) shared
capacity = (0=215040, 1=819200, 5=10485760)
```

As the preceding example shows, comments can be included in this file as long as they are preceded by a pound sign (#).

Configuring a shared drive differs from library to library. For more information on how to configure shared drives for one of these network-attached libraries, see the following man pages: ibm3494(7), sony(7), or stk(7).

By default, a cartridge in a shared drive can be idle for 60 seconds before being unloaded. To change this timing, specify the `shared_unload` directive in the `defaults.conf` file. For more information on this directive, see the `defaults.conf`(4) man page.

Load Notification

The ASM and ASM-QFS software requests cartridges to be loaded regularly to satisfy archiving and staging needs. If the request is for a cartridge that resides inside a library, the request is handled automatically. If the request is for a cartridge that resides outside the library, operator action is required.

If enabled, the `load_notify.sh`(1M) script sends email when a cartridge needs to be obtained from outside the library. The script itself resides in the following location:

```
/opt/SUNWsamfs/examples/load_notify.sh
```

Before it can be used, you must copy it to the following location:

```
/opt/SUNWsamfs/sbin/load_notify.sh
```

Certain directives must be in effect in the `defaults.conf` file. These are as follows:

- `exported_media=available`
- `attended=yes`

These directives are set by default. If the load notification capability is to be enabled, ensure that these directives have not been changed.

By default, the script sends email to `root`, but it can be edited to send email to another person, to dial a pager, or to provide some other means of notification.

Catalog Operations

A *library catalog* is the central repository of all information that the ASM or ASM-QFS environment needs for finding cartridges in an automated library. The library catalog file is a binary UFS-resident file that contains information about each slot in an automated library. The information in this file includes the one or more volume serial names (VSNs) associated with the cartridge stored in the slot, the capacity and space remaining on that cartridge, and flags indicating read-only, write-protect, recycling, and other status information for the cartridge.

The ASM and ASM-QFS environments treat catalogs differently depending on how the automated library is attached to the server, as follows:

- If the automated library is direct-attached, the library catalog is a one-to-one mapping between library catalog entries and physical slots in the automated library. The first entry in the library catalog is for the first slot in the automated library. When a cartridge is needed, the system consults the library catalog to determine which slot contains the VSN, and it issues a command to load the cartridge from that slot into a drive.
- If the automated library is network-attached, the library catalog is not a direct mapping to the slots. It is a list of the VSNs known to be present in the automated library. When a cartridge is requested, the system sends a request to the vendor's software to load the VSN into a drive. The vendor's software locates the VSN's storage slot.

There are several operations necessary to initialize and maintain your automated library in a ASM or ASM-QFS environment. The following sections show you how to build a library catalog, how to examine a library catalog, and how to add and remove cartridges from an automated library.

To Build a Library Catalog

When the `sam-catserverd` daemon starts, it checks to see that the library catalog file for each automated library is present. If an automated library's catalog file is not present, then a catalog must be created, as follows:

- For direct-attached and IBM 3494 automated libraries with barcoded media, ASM or ASM-QFS builds a catalog.
- For direct-attached and IBM 3494 libraries without barcoded media, ASM or ASM-QFS audits the library and then builds the catalog. The audit reveals the location of all cartridges in the automated library.
- For network-attached StorageTek, ADIC/Grau, Fujitsu, and Sony automated libraries, however, you must build the catalog by issuing a series of `import`(1M) commands or by issuing one `build_cat`(1M) command. Issuing a series of `import`(1M) commands is acceptable if there are not too many cartridges in the automated library. If you have many cartridges, however, you can supply a list of VSNs as an argument to the `build_cat`(1M) command. The `build_cat`(1M) command uses the VSN list to populate an initial catalog.

For more information on the `import`(1M) command, see the `import`(1M) man page. For more information on the `build_cat`(1M) command, see the `build_cat`(1M) man page.

To Examine a Library Catalog

The `dump_cat`(1M) command writes an automated library's catalog file in text format. This command has the following format:

```
dump_cat -o -V catalog_name
```

where:

- `-o` Lists cartridges that are no longer present in the library catalog. That is, the in-use flag is not set, but there is an entry present.
- `-V` Displays flags and label times as comments. It prints a line for each VSN, displaying the label time, last modification time, and load time. It displays the flags in the same format as the `samu`(1M) utility's `v` display.

The following is a sample `dump_cat` listing:

```
# dump_cat stk
# audit_time Wed Dec 31 18:00:00 1969
# version 350  count 100 mediatype sg
# Index  VSN  Barcode Type PTOC Access Capacity Space Status
Sector Label time  Eq  Slot Part
#
     0 004974 004974 sg  0xaaa6   27 20971520 15379712
0x7a010000 262144 0x386ccfad 300     0   0
     1 004971 004971 sg  0xe003   45 20971520 13631104
0x6a000000 262144 0x3793b707 300     1   0
     3 004973 004973 sg   0xb8c    7 20971520 20593152
0x6a000000 262144 0x38c94d17 300     3   0
     4 004975 004975 sg   0x660    5 20971520 20762624
0x6a000000 262144 0x3693a80d 300     4   0
     5 004977 004977 sg  0x5421   27 20971520 17243264
0x6a000000 262144 0x384eb998 300     5   0
     6 004970 004970 sg 0x1468e    1 20971520 20971520
0x6a000000 262144 0x38c94da1 300     6   0
```

Importing and Exporting Cartridges

The physical addition (import) and removal (export) of cartridges from an automated library can perform several functions. For example, you can replace cartridges, relocate disaster recovery tapes to off-site storage, and so on. Importing and exporting cartridges also updates the library catalog. The ASM or ASM-QFS system accomplishes these tasks using the `import`(1M) and `samexport`(1M) commands. You can also perform these tasks using `libmgr`(1M) or `robottool`(1M).

Note, however, that each automated library handles cartridge import and export differently due to system characteristics and the vendor-supplied software. For example, on the ACL 4/52 library, you need to issue a move command to move cartridges into the import/export unit before exporting cartridges from the automated library. Devices like the ADIC/Grau, StorageTek, and Fujitsu automated libraries import and export cartridges using their own utilities, so the `import`(1M) and `samexport`(1M) commands only update the library catalog entries used by the ASM and ASM-QFS systems.

For more information on importing and exporting cartridges, see "Importing and Exporting Cartridges (Systems With a Mailbox)" on page 39 and see "Importing and Exporting Cartridges (Systems Without a Mailbox)" on page 41.

Tracking Exported Media — The Historian

The ASM and ASM-QFS historians keep track of cartridges exported from an automated library or a manually mounted device. The historian acts like a virtual library, but it has no defined hardware devices. Like an automated library, it is configured in the `mcf` file, has a catalog that records entries for all cartridges associated with it, can import and export cartridges, and appears in `robottool`(1M) as another automated library.

The historian can be configured in the `mcf` file by using a device type of `hy`. If the historian is not configured in the `mcf` file, it is created as:

```
historian  n+1  hy  -  on  /var/opt/SUNWsamfs/catalog/
historian
```

In the preceding entry, `n+1` is the last Equipment Ordinal in the `mcf` file plus 1. If you want to use a different Equipment Ordinal or path name for the catalog, you need only to define the historian in the `mcf`.

The historian library catalog is initialized with 32 entries when the historian first starts. Make sure that the catalog resides on a file system large enough to hold the entire catalog. Your site might want to track existing ASM and ASM-QFS cartridges that have been exported from the library. In this case, you need to build a historian catalog from the existing cartridges as described in the `build_cat`(1M) man page.

The following two configuration directives in the `defaults.conf` file affect the behavior of the historian:

- If the `exported_media = unavailable` directive appears, then any cartridges exported from an automated library are flagged as unavailable to the historian. Requests for cartridges flagged as unavailable generate an EIO error.
- If the `attended = no` directive is set, it declares to the historian that no operator is available to handle load requests. Requests to load cartridges known to the historian, and not already loaded, generate an EIO error.

For more configuration information, see the `historian`(7) and `defaults.conf`(4) man pages.

Conventions

The procedures for performing the basic operations described in this chapter typically use the `samu`(1M) operator utility and the following commands:

- `tplabel`(1M)
- `odlabel`(1M)
- `auditslot`(1M)
- `cleandrive`(1M)
- `chmed`(1M)
- `import`(1M)
- `set_state`(1M)
- `samexport`(1M)

In many cases, however, there is more than one way to perform the task described. In addition to using `samu`(1M) and the commands, you can perform many of these tasks from within the graphical user interface (GUI) tools, which are `devicetool`(1M), `libmgr`(1M), and `robottool`(1M). For more information on the GUI tools, see "Graphical User Interface (GUI) Tools" on page 195.

Command Arguments

Many of the commands accept a common set of arguments. Table 10. shows these arguments.

Table 10. Command Arguments

Argument	Meaning
eq	The Equipment Ordinal of the device being addressed as defined in the `mcf` file. The device that is identified can be an automated library, a drive, or a file system.
slot	The number of a storage slot in an automated library as recognized in the library catalog.
partition	A side of a magneto-optical disk. The partition must be 1 or 2.
media_type	The media type. For a list of valid media types, see the `mcf`(4) man page.
vsn	The volume serial name assigned to the volume.

Some commands accept various combinations of arguments depending on your circumstances. For example, from the `samu`(1M) operator utility, the `load` command has the following two formats:

```
:load eq:slot
:load media_type.vsn
```

Note the following:

- The first form uses a colon (:) to separate *eq* and *slot*.
- The second form uses a period (.) to separate *media_type* and *vsn*.

Terminology

Certain terms used throughout this chapter might be new to you. Table 11. shows some of the most commonly used terms and their meanings.

Table 11. Terminology

Term	Meaning
Automated library	An automated device for storing tape and optical cartridges.
Cartridge	A tape or magneto-optical cartridge. A magneto-optical cartridge can contain one or more volumes or partitions.
Partition	An entire tape or one side of a magneto optical disk. A partition can contain only one volume.
Volume	A named area on a cartridge for storing data. A cartridge has one or more volumes. Double-sided cartridges have two volumes, one on each side. A volume serial name (VSN) identifies a volume.

■ Automated Library Operations

Several basic operations are essentially the same on all automated libraries. The following basic operations are explained in this section:

- “To Autoclean and Autoload” on page 26
- “To Start Removable Media Operations” on page 37
- “To Stop Removable Media Operations” on page 26
- “To Turn an Automated Library On” on page 27
- “To Turn an Automated Library Off” on page 28
- “To Load a Cartridge Into an Automated Library” on page 28

- “To Unload a Cartridge From a Drive” on page 29
- “Labeling a Cartridge” on page 30
- “To Audit a Volume” on page 32
- “To Audit an Automated Library (Direct-Attached Only)” on page 32
- “Using a Cleaning Cartridge” on page 33
- “To Clean a Tape Drive” on page 36
- “To Clear Media Errors” on page 36
- “To Remove a Stuck Cartridge From a Drive” on page 37

To Autoclean and Autoload

If your library is direct-attached and has autoloading or autocleaning capabilities, autocleaning and autoloading should be disabled.

If your library is network attached, the autocleaning and autoloading features should be enabled according to the recommendations of the library manufacturer.

Autocleaning and autoloading practices differ from manufacturer to manufacturer. If you are having trouble with this, see “Vendor-Specific Operational Procedures for Libraries” on page 45 to determine if specialized procedures are recommended for your equipment.

For information on how to enable or disable autocleaning and autoloading, see the documentation from your library’s manufacturer.

To Stop Removable Media Operations

It is possible to stop removable media operations and leave the ASM or ASM-QFS system mounted. You might do this, for example, if you want to manually manipulate cartridges in a library. When operations are resumed, pending stages are reissued and archiving is resumed.

To stop removable media operations, enter the following commands:

```
samcmd idle eq
samd stop
```

where:

eq The Equipment Ordinal of the equipment being addressed as defined in the `mcf` file. To idle the drives, enter a `samcmd idle` *eq* command for each *eq* configured in your `mcf` file.

You can also idle the drives by using the `samu`(1M) operator utility or by using either the `robottool`(1M) or `libmgr`(1M) GUI tools. For more information on the `samcmd`(1M) command, see the `samcmd`(1M) man page. For more information on the `samd`(1M) command, see the `samd`(1M) man page.

Note: The drives in your ASM or ASM-QFS environment should be idled prior to issuing the `samd stop` command. This allows the archiver, stager, and other processes to complete current tasks. Failure to issue the `samd stop` command can cause unexpected results when archiving, staging, and other activities are resumed.

To Start Removable Media Operations

Typically, removable media operations commence when a ASM or ASM-QFS file system is mounted. To start removable media operations manually, without mounting any file systems, enter the following `samd`(1M) command:

```
# samd start
```

If removable media operations are already running when the preceding command is entered, the following message is generated:

```
SAM-FS sam-initd daemon already running
```

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

To Turn an Automated Library On

When a library is in the `on` state, it is under the control of the ASM or ASM-QFS system and can proceed with general operations. When you turn on a library, the ASM or ASM-QFS software performs the following actions:

- Queries the device regarding its internal state. It discovers where tapes are, whether or not barcodes are used, and so on.
- Updates the catalog and other internal structures.

The `samu`(1M) utility's `s` display shows the state of an automated library. To turn an automated library on, use the `samu`(1M) utility's `:on` command. This command has the following format:

```
:on eq
```

where:

eq The Equipment Ordinal of the automated library being addressed as defined in the `mcf` file.

This task can also be performed by using the GUI tools `robottool`(1M) and `libmgr`(1M). For more information on these tools, see “Graphical User Interface (GUI) Tools” on page 195 or see the `robottool`(1M) and `libmgr`(1M) man pages.

To Turn an Automated Library Off

Placing a library in the `off` state stops I/O operations and removes the automated library from the ASM or ASM-QFS control. No automatic movement of cartridges occurs. Note that the drives in the automated library remain in the `on` state. You may want to turn an automated library off to perform the following tasks:

- To stop ASM or ASM-QFS operations for this automated library only.
- To power down the automated library.

To turn an automated library to `off`, use the `samu`(1M) utility’s `:off` command. This command has the following format:

```
:off eq
```

where:

eq	The Equipment Ordinal of the equipment being addressed as defined in the `mcf` file.

This task can also be performed by using the GUI tools `robottool`(1M) and `libmgr`(1M). For more information on these tools, see “Graphical User Interface (GUI) Tools” on page 195 or see the `robottool`(1M) and `libmgr`(1M) man pages.

To Load a Cartridge Into an Automated Library

Loading a cartridge into a drive occurs automatically when a VSN is requested for archiving or staging. Loading refers to moving a cartridge from a storage slot to a drive and making it ready.

To manually load a cartridge, use the `samu`(1M) utility’s `:load` command. This command can be used even if the drive is in `unavail` status. This command has the following two possible formats:

```
:load eq:slot[:partition]
:load media_type.vsn
```

where:

eq	The Equipment Ordinal of the drive being addressed as defined in the `mcf` file.
slot	The number of a storage slot in an automated library as recognized in the library catalog.
media_type	The media type. For a list of valid media types, see the `mcf`(4) man page.
partition	A side of a magneto-optical disk. The partition must be 1 or 2. This argument is not applicable to tape cartridges.
vsn	The volume serial name assigned to the volume.

This task can also be performed by using the GUI tools `robottool`(1M) and `libmgr`(1M). For more information on these tools, see "Graphical User Interface (GUI) Tools" on page 195 or see the `robottool`(1M) and `libmgr`(1M) man pages.

When you manually load a cartridge, it is generally loaded in the next available drive in the library. If you want to make a drive unavailable for this purpose, use the `samu`(1M) utility's `:unavail` command or change the state of the device using `devicetool`(1M). You might do this, for example, during a disaster recovery operation or to analyze a tape.

To Unload a Cartridge From a Drive

Unloading a cartridge occurs automatically when a volume is no longer needed. You can also manually unload a drive. Unloading refers to removing a cartridge from a drive.

To manually unload a drive, use the `samu`(1M) utility's `:unload` command. This command can be used even if the drive is in `unavail` status. This command has the following format:

```
:unload eq
```

where:

eq	The Equipment Ordinal of the drive being addressed as defined in the `mcf` file.

This task can also be performed by using the GUI tools `robottool`(1M) and `libmgr`(1M). For more information on these tools, see "Graphical User Interface (GUI) Tools" on page 195 or see the `robottool`(1M) and `libmgr`(1M) man pages.

Labeling a Cartridge

If you are not using an automated library with a barcode reader, you must label all cartridges before using the ASM or ASM-QFS system.

If your library uses barcodes, `labels = barcodes` is set by default, and the result is that the first six characters are used for the VSN.

If your library uses barcodes, and you want the last six characters to become the VSN for the cartridge, edit the `/etc/opt/SUNWsamfs/defaults.conf` file and include the following line:

```
labels = barcodes_low
```

When a barcoded cartridge is loaded for a write operation, a label is written on the cartridge before the write begins. The cartridge must be write enabled, unlabeled, and have a readable barcode.

The procedure for labeling a cartridge differs depending on whether you are labeling a tape or optical cartridge. The following two sections describe these procedures.

Caution: Labeling and relabeling a cartridge makes the data on the cartridge unaccessible to the ASM and ASM-QFS software.

To Label or Relabel a Tape

The following `tplabel`(1M) command line format shows the options most commonly used when labeling or relabeling a tape:

```
tplabel [ -new | -old vsn ] -vsn vsn eq:slot
```

where:

vsn	A volume serial name. If you are relabeling, the new VSN name can be identical to the old VSN name.
eq	The Equipment Ordinal of the automated library or manually loaded drive being addressed as defined in the `mcf` file.
slot	The number of a storage slot in an automated library as recognized in the library catalog. This argument is not applicable for manually loaded drives.

To label a new tape, use the `tplabel` command with the following options:

```
tplabel -new -vsn vsn eq:slot
```

To relabel an existing tape, use the `tplabel` command with the following options:

```
tplabel -old vsn -new -vsn vsn eq:slot
```

After issuing the command to label or relabel a tape, the tape is loaded and positioned, and the tape label is written. For more information on the `tplabel`(1M) command, see the `tplabel`(1M) man page.

This task can also be performed by using the GUI tools `robottool`(1M) and `libmgr`(1M). For more information on these tools, see "Graphical User Interface (GUI) Tools" on page 195 or see the `robottool`(1M) and `libmgr`(1M) man pages.

To Label or Relabel an Optical Disk

The following `odlabel`(1M) command line format shows the options most commonly used when labeling or relabeling an optical disk:

```
odlabel [ -new | -old vsn ] -vsn vsn eq:slot:partition
```

where:

vsn	A volume serial name. If you are relabeling, the new VSN name can be identical to the old VSN name.
eq	The Equipment Ordinal of the automated library or manually loaded drive being addressed as defined in the `mcf` file.
slot	The number of a storage slot in an automated library as recognized in the library catalog. This argument is not applicable to manually loaded drives.
partition	A side of a magneto-optical disk. The partition must be `1` or `2`. This argument is not applicable to tape cartridges.

To label a new optical disk, use the `odlabel`(1M) command with the following options:

```
odlabel -new -vsn vsn eq:slot:partition
```

To relabel an existing optical disk, use the `odlabel`(1M) command with the following options:

```
odlabel -old vsn -vsn vsn eq:slot:partition
```

After issuing the command to label or relabel an optical disk, the optical disk is loaded and positioned, and the optical disk label is written. For more information on the `odlabel`(1M) command, see the `odlabel`(1M) man page.

This task can also be performed by using the GUI tools `robottool`(1M) and `libmgr`(1M). For more information on these tools, see “Basic Operations” on page 13 or see the `robottool`(1M) and `libmgr`(1M) man pages.

To Audit a Volume

Occasionally the reported space remaining on a tape or optical cartridge might need to be updated in the library catalog. The `auditslot`(1M) command loads the cartridge containing the volume, reads the label, and updates the library catalog entry for the slot.

This command has the following format:

```
auditslot [-e] eq:slot[:partition]
```

where:

`-e`	If the `-e` option is specified, the remaining space is updated. Otherwise, it is not changed.
eq	The Equipment Ordinal of the automated library or manually loaded drive being addressed as defined in the `mcf` file.
slot	The number of a storage slot in an automated library as recognized in the library catalog. This argument is not applicable to manually loaded drives.
partition	A side of a magneto optical disk. The partition must be `1` or `2`. This argument is not applicable to tape cartridges.

For more information on the `auditslot`(1M) command, see the `auditslot`(1M) man page.

This task can also be performed by using the `samu`(1M) utility’s `:audit` command or by using the GUI tools `robottool`(1M) and `libmgr`(1M). For more information on these tools, see “Graphical User Interface (GUI) Tools” on page 195; “Using the samu(1M) Operator Utility” on page 223; or one of the following man pages: `samu`(1M), `robottool`(1M), or `libmgr`(1M).

To Audit an Automated Library (Direct-Attached Only)

Note: This task cannot be performed on a network-attached automated library.

A full audit loads each cartridge into a drive, reads the label, and updates the library catalog. A library should be audited in the following situations:

- After moving cartridges in the automated library without using ASM or ASM-QFS commands.
- If you are in doubt about the status of the library catalog and would like to update it (for example, after a power outage).
- If you have added, removed, or moved cartridges in an automated library that has no mailbox.

To perform a full audit on an automated library, use the `samu`(1M) utility's `:audit` command. This command has the following format:

```
:audit eq
```

where:

eq The Equipment Ordinal of the drive being addressed as defined in the `mcf` file.

This task can also be performed by using the `samu`(1M) utility's `:audit` command or by using the GUI tools `robottool`(1M) and `libmgr`(1M). For more information on these tools, see "Graphical User Interface (GUI) Tools" on page 195; "Using the samu(1M) Operator Utility" on page 223; or one of the following man pages: `samu`(1M), `robottool`(1M), or `libmgr`(1M).

Using a Cleaning Cartridge

The ASM and ASM-QFS systems allow you to import a cleaning cartridge to clean tape drives. This procedure differs depending on whether or not the cleaning cartridge is barcoded. The following sections explain various aspects of using a cleaning cartridge.

Cleaning practices differ from manufacturer to manufacturer. If you are having trouble with this, see "Vendor-Specific Operational Procedures for Libraries" on page 45 to determine if specialized procedures are recommended for your equipment.

Note: This task cannot be performed on a network-attached automated library.

To Reset the Number of Cleaning Cycles

Cleaning tapes are useful only for a limited number of cleaning cycles. The number of remaining cycles can be viewed with `samu`(1M) utility's `:v` display or from within either of the GUI tools `robottool`(1M) or `libmgr`(1M). For more information on these tools, see "Graphical User Interface (GUI) Tools" on page 195; “Using the samu(1M) Operator Utility” on page 223; or see one of the following man pages: `samu`(1M), `robottool`(1M), or `libmgr`(1M).

The ASM and ASM-QFS systems track the number of cleaning cycles used for each cleaning tape and eject the tape when the remaining cycles equal zero. For example, a DLT cleaning tape has 20 cycles, and an Exabyte cleaning tape has 10 cycles. Each time a cleaning tape is imported, the cleaning cycle is reset to the highest number of cycles for that type of tape.

If automatic cleaning is available on your system but all cleaning tapes in the automated library have a count of zero, the drive is set to off and a message is issued in the ASM or ASM-QFS log. You can reset a cleaning tape with a count of zero by using the `chmed`(1M) command. This command has the following format:

```
chmed -count count media_type.vsn
```

where:

count	The number of cleaning cycles to which you want the cleaning tape reset.
media_type	The media type. For a list of valid media types, see the `mcf`(4) man page.
vsn	The volume serial name assigned to the volume.

To Use Cleaning Cartridges With Barcodes

If the cleaning cartridge is barcoded, you can import it using the `import`(1M) command. The barcode must be `CLEAN`, or it must start with the letters `CLN`.

The `import`(1M) command has the following format:

```
import eq
```

where:

eq	The Equipment Ordinal of the drive being addressed as defined in the `mcf` file.

The ASM and ASM-QFS systems move the cartridges from the mailbox to a storage slot and update the library catalog for each cartridge.

For example, the following command imports a cleaning tape into the automated library that is numbered 50 in your mcf file:

```
# import 50
```

After this command is issued, the cleaning media flag is set, and the access count is set to the appropriate number of cleaning cycles, based on the media type. Each time the media is used to clean a drive, the access count is decremented.

This procedure can also be performed from within samu(1M) or from within one of the GUI tools robottool(1M) or libmgr(1M). For more information on these tools, see the samu(1M), robotool(1M), or libmgr(1M) man pages.

To Use Cleaning Cartridges Without Barcodes

If the cartridge is not barcoded, you must import it first. It does not become marked as a cleaning cartridge. Perform the following steps:

1. Import the cartridge using the import(1M) command.

The import(1M) command has the following format:

```
import eq
```

where:

eq The Equipment Ordinal of the drive being addressed as defined in the mcf file.

2. Use the chmed(1M) command to change the type to a cleaning cartridge.

You must know the Equipment Ordinal of the automated library and the slot into which the cleaning cartridge is loaded.

In the following example command line, the automated library is Equipment Ordinal 50 and the cleaning cartridge is in slot 77:

```
# chmed +C 50:77
```

The preceding command changes the cartridge type to that of a cleaning cartridge.

3. Use chmed(1M) again to set the cleaning cycle count.

The following example command sets the count on the cartridge used in the preceding step:

```
# chmed -count 20 50:77
```

For more information on the chmed(1M) command, see the chmed(1M) man page.

To Clean a Tape Drive

Note: The ASM and ASM-QFS systems do not support automatic cleaning on network-attached libraries. You should use the vendor's library manager software for automatic cleaning.

The ASM and ASM-QFS environments support the use of cleaning tapes if cleaning tapes are supported by the hardware. If a tape drive requests cleaning, the system automatically loads a cleaning tape.

If your system uses barcoded labels, cleaning tapes must have a VSN of CLEAN or a VSN starting with the letters CLN in the barcode label. Alternatively, you can use the chmed(1M) command to mark a VSN as a cleaning tape and set the count. Multiple cleaning tapes are allowed in a system.

Note: Certain drive errors can result in cleaning cartridges being loaded repeatedly until all cleaning cycles are consumed. You can prevent this situation by using the chmed(1M) command to limit the number of cleaning cycles on cleaning cartridges. For example:

```
# chmed -count 20 50:77
```

When automatic cleaning is not available and the system uses barcodes, you can use the cleandrive(1M) command to manually request that a drive be cleaned. This command has the following format:

```
cleandrive eq
```

where:

eq The Equipment Ordinal of the drive being addressed as defined in the mcf file. This is the drive to be loaded with the cleaning cartridge.

To Clear Media Errors

When a hardware or software error is encountered on a cartridge, the ASM system sets the media error flag in the VSN catalog. On any given cartridge that generates a media error signal, the chmed(1M) command can be used to clear the error, and you can attempt to use the cartridge. The `media error` flag is displayed in the `samu`(1M) utility's `v` display, in the `robottool`(1M) VSN catalog panel, and in the `libmgr`(1M) VSN display.

1. Issue the `chmed`(1M) command to clear the `media error` flag.

The following format shows the `chmed`(1M) command options to use to clear the `media error` flag:

```
chmed -E media_type.vsn
```

where:

media_type	The media type. For a list of valid media types, see the `mcf`(4) man page.
vsn	The volume serial name assigned to the volume.

2. Issue the `auditslot`(1M) command to update the space remaining information.

   The following format shows the `auditslot`(1M) command option to use to accomplish this:

```
auditslot -e
```

To Remove a Stuck Cartridge From a Drive

If a cartridge becomes stuck in a drive, follow these steps.

1. Turn off the drives in the automated library.

To turn off a drive, use the `samu`(1M) utility's `:off` command. This command has the following format:

```
:off eq
```

where:

eq	The Equipment Ordinal of the equipment being addressed as defined in the `mcf` file.

This task can also be performed by using the GUI tools `robottool`(1M) and `libmgr`(1M). For more information on these tools, see “Graphical User Interface (GUI) Tools” on page 195 or see the `robottool`(1M) or `libmgr`(1M) man pages.

2. Turn off the automated library.

To turn off a library, use the `samu`(1M) utility's `:off` command. This command has the following format:

```
:off eq
```

where:

eq The Equipment Ordinal of the equipment being addressed as defined in the `mcf` file.

This task can also be performed by using the GUI tools `robottool`(1M) and `libmgr`(1M). For more information on these tools, see "Graphical User Interface (GUI) Tools" on page 195 or see the `robottool`(1M) or `libmgr`(1M) man pages.

3. Physically remove the cartridge from the drive.

   Make sure you do not damage either the cartridge or the drive.

4. Turn on the automated library and the drive.

To turn on a library or a drive, use the `samu`(1M) utility's `:on` command. This command has the following format:

```
:on eq
```

where:

eq The Equipment Ordinal of the equipment being addressed as defined in the `mcf` file.

   If the automated library performs an audit when it is turned on, you are done. If it does not, perform the next step.

5. If you put the cartridge back into its storage slot, adjust the library catalog to set the occupied flag for the damaged tape by using the `chmed`(1M) command.

   This command has the following format:

```
chmed +o eq:slot
```

where:

eq	The Equipment Ordinal of the automated library or drive being addressed as defined in the `mcf` file.
slot	The number of a storage slot in a library as recognized in the library catalog. This argument is not applicable for manually loaded drives.

For more information on the `chmed`(1M) command, see the `chmed`(1M) man page.

If you keep the cartridge out, and you want to put it back in later, you must import the cartridge into the automated library.

Importing and Exporting Cartridges (Systems With a Mailbox)

This section explains how to import and export media for direct-attached automated libraries that use a mailbox. These libraries include, for example, the StorageTek 9714, StorageTek 9710, StorageTek 9740, and ADIC Scalar series.

Note: If you have a network-attached library, see “Vendor-Specific Operational Procedures for Libraries” on page 45 for information on importing and exporting cartridges.

A *mailbox* is an area in an automated library for adding and removing cartridges from the automated library. The `import`(1M) command moves a cartridge from the mailbox to a storage slot. The `samexport`(1M) command moves the cartridge from a storage slot to the mailbox. For most libraries, if a cartridge is present in the mailbox at the time that the ASM and ASM-QFS software is started, the software imports the cartridge automatically upon startup.

Importing and exporting practices differ from manufacturer to manufacturer. If you are having trouble with this, see “Vendor-Specific Operational Procedures for Libraries” on page 45 to determine if specialized procedures are recommended for your equipment.

If your system does not have a mailbox, the operations do not apply to your system. You should read “Importing and Exporting Cartridges (Systems Without a Mailbox)” on page 41.

To Import a Cartridge Using a Mailbox

To import cartridges into an automated library that uses a mailbox, follow these steps.

1. Open the mailbox using the manufacturer's suggested operation.

   There is usually a button near the mailbox. Sometimes the mailbox is a one-slot mailbox referred to as a *mail slot* in the vendor's documentation.

2. Manually place the cartridge into the mailbox.
3. Close the mailbox.
4. Use the `import`(1M) command to import the cartridge.

   This command has the following format:

```
import eq
```

where:

eq	The Equipment Ordinal of the automated library being addressed as defined in the `mcf` file.

The system moves the cartridge from the mailbox to a storage slot and updates the library catalog for each cartridge.

You can also perform this step from within `samu`(1M) or from within one of the GUI tools `robottool`(1M) or `libmgr`(1M). For more information on these tools, see the `samu`(1M), `robotool`(1M), or `libmgr`(1M) man pages.

To Export a Cartridge Using a Mailbox

This procedure moves a cartridge from a storage slot to a mailbox or mail slot. To export (eject) cartridges from a library that uses a mailbox, follow these steps.

1. Use the `samexport`(1M) command, in one of the following formats, to move a cartridge from a storage slot to the mailbox:

```
samexport eq:slot
samexport media_type.vsn
```

where:

eq	The Equipment Ordinal of the automated library being addressed as defined in the `mcf` file.

slot	The number of a storage slot in an automated library as recognized in the library catalog.
media_type	The media type. For a list of valid media types, see the `mcf`(4) man page.
vsn	The volume serial name assigned to the volume.

You can also perform this step from within the `samu`(1M) utility or by using one of the GUI tools `robottool`(1M) or `libmgr`(1M). For more information on these tools, see “Graphical User Interface (GUI) Tools” on page 195; “Using the samu(1M) Operator Utility” on page 223; or one of the following man pages: `samu`(1M), `robottool`(1M), or `libmgr`(1M).

2. Open the mailbox or mail slot using the manufacturer’s suggested operation.

   There is usually a button near the mailbox.

Importing and Exporting Cartridges (Systems Without a Mailbox)

This section explains how to manipulate cartridges if you have an automated library that does not use a mailbox.

To Import a Cartridge Without a Mailbox

1. Invoke the `samu`(1M) utility, and enter the `:unload` command.

   This command has the following format:

```
:unload eq
```

where:

eq	The Equipment Ordinal of the device being addressed as defined in the `mcf` file.

Wait until the system completes its current task, sets the status to off, and transfers the current active catalog to the historian.

2. Unlock and open the door to the automated library.
3. Load cartridges into the available slots.
4. Close and lock the door to the automated library.

The automated library reinitializes and scans the cartridges in the library. The ASM or ASM-QFS software updates the library catalog by adding the VSNs of the imported cartridges to the catalog. The automated library state is set to `on`.

To Export a Cartridge Without a Mailbox

1. Invoke the `samu`(1M) utility, and enter the `:unload` command.

   This command has the following format:

```
:unload eq
```

where:

eq	The Equipment Ordinal of the device being addressed as defined in the `mcf` file.

Wait until the system completes its current task, sets the status to `off`, and transfers the current active library catalog to the historian file.

2. Unlock and open the door to the automated library.
3. Remove the cartridges from their respective slots.
4. Close and lock the door to the automated library.

The automated library reinitializes and scans the cartridges in the automated library. The system updates the library catalog with the VSNs of the cartridges currently in library slots. The VSNs of the removed cartridges are removed from the library catalog and are now recorded only in the historian file. The automated library state is set to `on`.

■ Manually Loaded Drive Operations

This section describes operations that differ if you have a manually loaded, standalone drive and not an automated library. Each manually loaded drive has its own one-slot library catalog.

To Load a Cartridge

- To load a cartridge into a manually loaded device, place the cartridge in the drive according to the manufacturer’s instructions.

The ASM and ASM-QFS systems recognize that the cartridge is loaded, read the label, and update the manual, one-slot catalog. No further action is necessary.

To Unload a Cartridge

- Use `samu`(1M) utility's `:idle` command to idle the drive.

  The `:idle` command ensures that no archive or stage processes are active. The format of this command is as follows:

```
:idle eq
```

where:

eq	The Equipment Ordinal of the drive being addressed as defined in the `mcf` file.

The drive switches from `idle` to `off` when all I/O activity is complete, and the tape ejects.

If this is a tape, the tape rewinds, and the cartridge is ready to be removed. An optical cartridge ejects automatically. See the manufacturer's instructions for removing the specific cartridge.

You can perform this entire procedure from within the GUI tools `libmgr`(1M) or `devicetool`(1M). For more information on these tools, see "Graphical User Interface (GUI) Tools" on page 195 or see the `libmgr`(1M) or `devicetool`(1M) man pages.

To View a Library Catalog

- Use the `samu`(1M) utility's `:v` command.

The format of this command is as follows:

```
:v eq
```

where:

eq	The Equipment Ordinal of the drive being addressed as defined in the `mcf` file.

Vendor-Specific Operational Procedures for Libraries

3

Many libraries can be used with ASM or ASM-QFS using the operational procedures described in “Basic Operations” on page 13. Some libraries, however, have vendor-specific operational procedures, and these are described in this chapter.

Note: The ASM and ASM-QFS software interoperates with automated libraries from many manufacturers. Consult the README file distributed with this release for information pertinent to library model numbers, firmware levels, and other compatability information.

This chapter contains the following topics:

ADIC/Grau Automated Libraries

The ADIC/Grau automated library operates within ASM and ASM-QFS environments through the `grauaci` interface. The `grauaci` interface is the interface between the ASM and ASM-QFS software and the GRAU ABBA library. This interface uses the DAS/ACI 3.02 interface supplied by ADIC/GRAU. For more information on DAS/ACI, see the *DAS/ACI 3.02 Interfacing Guide* and the *DAS Administration Guide*, both of which are available from ADIC/Grau.

Configuration

The ASM or ASM-QFS configuration process should not be attempted until the ADIC/Grau automated library is operational and the ABBA library is operating on the DAS server. In the DAS configuration file for this client, both the `avc` (avoid volume contention) and the `dismount` parameters should be set to `true`.

The following sections describe other aspects of configuration that are specific to ADIC/Grau automated library operations.

The mcf File

When creating the `mcf` file to include one or more ADIC/Grau automated libraries, the `Equipment Identifier` field in the `mcf` file must be the full path name to the parameters file used by the `grauaci` interface.

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

ADIC/Grau Parameters File

The ADIC/Grau parameters file consists of a list of *keyword* = *value* parameter lines. The various *keyword* values identify the ADIC/Grau automated libraries, the drives associated with the libraries, and the server name. All *keyword* and *value* entries are case sensitive, so they must be entered exactly as used in the DAS configuration file and in the ASM or ASM-QFS `mcf` file. The following types of *keyword* = *value* parameters must appear in the ADIC/Grau parameters file:

- `client =` *client_id*. For *client_id*, specify the name of the client as defined in the DAS configuration file. This is a required parameter.
- `server =` *server_id*. For *server_id*, specify the host name of the server running the DAS server code. This is a required parameter.
- `acidrive` *drive_id* = *path*. For *drive_id*, specify the name of the drive as configured in the DAS configuration file. For *path*, specify the path to the drive as entered in the `Equipment Identifier` field of the ASM or ASM-QFS `mcf` file. There must be one `acidrive` line for every drive assigned to the client.

Comments can appear anywhere on any line, but they must be preceded by a pound sign (#). Characters to the right of the pound sign are ignored.

If the ABBA library contains various media types, then there is a media changer for each media type. Each media changer has a unique client name in the DAS configuration, a unique library catalog, and a unique parameters file.

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

Example. This example shows a ASM `mcf` file and two ADIC/Grau parameters files. The `mcf` file is as follows:

```
#
# Sample mcf file entries for a GRAU library - DLT
#
/etc/opt/SUNWsamfs/grau50  50   gr   gr50  -  /var/opt/
SUNWsamfs/catalog/gr50
/dev/rmt/0cbn              51   lt   gr50  -  /dev/samst/
c2t5u0
/dev/rmt/1cbn              52   lt   gr50  -  /dev/samst/
c2t6u0
#
# Sample mcf file entries for a GRAU library - HP optical
#
/etc/opt/SUNWsamfs/grau60   60   gr  gr60  -  /var/opt/
SUNWsamfs/catalog/gr60
/dev/samst/c1t1u0           61   od  gr60  -
```

The following are the two parameters files referenced by the preceding `mcf` file. These files define one ADIC/Grau automated library supporting DLT tape and one ADIC/Grau automated library supporting a Hewlett Packard optical drive.

Figure 2. shows the first parameters file.

Figure 2. File `/etc/opt/SUNWsamfs/grau50`

```
# This is file: /etc/opt/SUNWsamfs/grau50
#
client = DASclient
server = DAS-server
#
# the name "drive1" is from the DAS configuration file
#
acidrive drive1 = /dev/rmt/0cbn     # a comment
#
# the name "drive2" is from the DAS configuration file
#
acidrive drive2 = /dev/rmt/1cbn     # a comment
```

Figure 3. shows the second parameters file.

Figure 3. File `/etc/fs/samfs/grau60`

```
# This is file: /etc/opt/SUNWsamfs/grau60
#
client = DASclient
server = DAS-server
```

Figure 3. File /etc/fs/samfs/grau60

```
acidrive DH03 = /dev/samst/c1t1u0
#
# the name "DH03" is from the DAS configuration file
```

Building a Catalog

The ASM and ASM-QFS systems automatically build a library catalog for an ADIC/Grau automated library. However, you must populate the library catalog. There are several ways of doing this. The appropriate method depends on the size of the catalog you need. The methods are as follows:

- **Method 1.** Create a catalog with existing VSN entries. (Note that this method works only for tapes. It does not work for barcoded optical media.) You can build a catalog that contains entries for many tapes by using the `build_cat`(1M) command. As input to `build_cat`(1M), you need to create a file that contains the slot number, VSN, barcode, and media type. For example, file `input_vsns` follows:

```
0 TAPE01  TAPE01  lt
1 TAPE02  TAPE02  lt
2 TAPE03  TAPE03  lt
```

The `input_vsns` file can be used as input to the `build_cat`(1M) command, as follows:

```
# build_cat input_vsns /var/opt/SUNWsamfs/grau50cat
```

- **Method 2.** Create a null catalog and import VSN entries. You can create an empty catalog and populate it. To create a catalog for 1000 slots, use the `build_cat`(1M) command, as follows:

```
# build_cat -s 1000 /dev/null /var/opt/SUNWsamfs/
catalog/grau50cat
```

Use the `import`(1M) command to add VSNs to this catalog, as follows:

```
# import -v TAPE01 50
```

- **Method 3.** Use the default catalog and import VSN entries. If a catalog path name is not specified in the `mcf` file, a default catalog is created in `/var/opt/SUNWsamfs/catalog/`*family_set_name* when ASM or

ASM-QFS is initialized. Following initialization, you must import VSN entries to this catalog. Use the import(1M) command, as follows:

```
# import -v TAPE01 50
```

In the preceding import(1M) command, 50 is the Equipment Ordinal of the automated library as specified in the mcf file.

The audit(1M) command is not supported for ADIC/Grau automated libraries.

Operations

“Basic Operations” on page 13 describes most of the basic operations that can be performed with an automated library. These operations include labeling cartridges, loading cartridges, and so on. From within the ASM or ASM-QFS environment, most basic operations are the same on most automated libraries. However, some basic operations for ADIC/Grau automated libraries differ from those described in “Basic Operations” on page 13 in the area of importing and exporting cartridges.

Because you use vendor-supplied utilities to physically add and remove cartridges in the ADIC/Grau automated library, the ASM import(1M) and samexport(1M) commands and the ASM libmgr(1M) import and export menus affect only the library catalog.

You can also perform the import and export procedures using samu(1M), robottool(1M), or libmgr(1M). For more information on these tools, see the samu(1M), robottool(1M), or libmgr(1M) man pages. For more information on the import(1M) and samexport(1M) commands, see the import(1M) and samexport(1M) man pages.

The import and export process is explained in the following sections.

To Import a Cartridge

To import a cartridge, perform the following steps.

1. Use ADIC/Grau commands to physically move the cartridge into the library.

2. Use the ASM and ASM-QFS import(1M) command to update the library catalog.

   The syntax for the import(1M) command is as follows:

```
import -v volser eq
```

where:

volser	The *volser* to be added. The `grauaci` interface verifies that the ADIC/Grau automated library has the *volser* information before updating the library catalog with the new entry.
eq	The Equipment Ordinal of the device being addressed as defined in the `mcf` file.

To Export a Cartridge

To export a cartridge, perform the following steps.

1. Use the ASM and ASM-QFS `samexport`(1M) command to remove the entry from the library catalog.

   The syntax for the `samexport`(1M) command is one of the following:

```
samexport eq:slot
samexport media_type.vsn
```

where:

eq	The Equipment Ordinal of the device being addressed as defined in the `mcf` file.
slot	The number of a storage slot in an automated library as recognized in the library catalog.
media_type	The media type. For a list of valid media types, see the `mcf`(4) man page.
vsn	The volume serial name assigned to the volume.

The `samexport`(1M) command updates the library catalog as each VSN is exported, and it moves the library catalog entry for each VSN from the library catalog to the historian.

2. Use ADIC/Grau commands to physically move the cartridge out of the library.

Diagnostic Information

Diagnostic information that can be useful when troubleshooting is located in the following directory:

```
/var/opt/SUNWsamfs/.grau
```

The system creates files in this directory that are named `graulog-`*eq*, where *eq* is the Equipment Ordinal as defined in the `mcf` file. For more information on this, see the `grauaci`(7) and the `mcf`(4) man pages.

ADIC Scalar Series Libraries

The ADIC Scalar 100, Scalar 224, and Scalar 448 series automated libraries contain either 2 or 4 DLT drives with a capacity of up to 48 DLT tapes. These automated libraries optionally use a mailbox and barcode readers.

Configuration

These ADIC libraries should not be configured with autoclean or autoload when running ASM or ASM-QFS software. Autoload can be used during the initial loading of cartridges as long as the system is not running. Remember to disable autoload when the ASM or ASM-QFS system is running.

Operations

Use the utilities described in “Basic Operations” on page 13 to perform importing, exporting, and other tasks.

Ampex Automated Libraries

Ampex automated libraries (such as the DST 812) use Ampex D2 tapes and drives. This section describes the configuration and initialization of these systems.

The Ampex automated libraries are configured like other direct-attached libraries. The tape drivers and runtime library software are supplied by Ampex and must be installed before installing the ASM or ASM-QFS software. The ASM and ASM-QFS systems require version 3.4, or later, of the Ampex DST Tape Device Driver.

To Configure an Ampex Library

To configure an Ampex library for use with a ASM or ASM-QFS system, you need to properly identify the system in the `/etc/opt/SUNWsamfs/mcf` file. Figure 4. shows the `mcf` file entries that configure a single D2 tape drive with an automated library.

Figure 4. Ampex Library Entries in an `mcf` File

```
# Equipment         Eq   Eq   Family   Dev   Additional
# Identifier        Ord  Ty   Set      St    Parameters
```

Figure 4. Ampex Library Entries in an `mcf` File

```
#
/dev/samst/c5t6u0   55   rb  am55    on
/dev/rdst6,1        56   d2  am55    on   /dev/rdst6,7
```

The following procedure shows how to configure two automated libraries.

1. Determine the `/dev/samst` symbolic links that point to the `/devices` files with the same hardware paths as shown in the `/var/adm/messages` files.

   In this example, `/dev/samst/c5t6u0`.
2. Determine the Equipment Ordinal.

   In this example, it is `55`.
3. Define the Equipment Type.

   In this example, it is `rb`.
4. Associate the automated library and drive using a common Family Set name.

   In this example, it is `am55`.
5. Set the device state to `on`.
6. Modify `/usr/kernel/drv/dst.conf`.

   The `DST_ZERO_ON_EW` device driver bit must be set as follows:

```
(set dst_dev_options = 0x00004001)
```

7. After you modify the `/usr/kernel/drv/dst.conf` file, you must reboot.

The Equipment Type for the Ampex D2 drive must be `d2`. Do not use the generic Equipment Identifier, `tp`, for these systems. The family set name is the same as the one chosen for the library (`am55`).

To Format a D2 Tape

It is not necessary to use Ampex-supplied commands to format tapes because the ASM system automatically formats them.

The ASM and ASM-QFS systems format tapes using the `tplabel`(1M) command with the `-erase` option. During formatting, the system writes several system zones to the tape. These zones contain the volume ID, which must be identical to the tape's ANSI-standard cartridge label. The zones allow the system to identify the loaded tape without rewinding it to the beginning to read the ANSI-standard label.

The ASM and ASM-QFS systems can relabel formatted D2 tapes by using the -erase option from the tplabel(1M) command or by checking the erase option in the label pop-up menu when using devicetool(1M). Using the -erase option reformats the tape. The new VSN is used as the vol_id in the system zones. If the -erase option is not selected, the relabel can still be completed, but you must use the tape's previous label in order to keep the system zone vol_id and the ANSI label synchronized.

For more information, see the tplabel(1M) or devicetool(1M) man pages.

Operations

When the ASM or ASM-QFS software takes control of the system, the buttons on the front panel of the automated library are not disabled as they are for other automated libraries. Use the utilities described in "Basic Operations" on page 13 to perform importing and exporting and to perform other tasks.

Caution: Do not use the buttons on the front of the Ampex libraries to load and unload tapes. Using the buttons causes undefined behavior in ASM and ASM-QFS environments and in the automated library.

■ Fujitsu LMF Automated Libraries

The Fujitsu LMF automated library operates in ASM and ASM-QFS environments through the LMF interface supplied by Fujitsu. The fujitsulmf interface is the interface between the ASM or ASM-QFS software and the Fujitsu LMF automated libraries. For more information on LMF, see the *LMF MTL Server/Client User's Guide* or the *LMF SAM-FS Linkage Operations Guide*. Both of these publications are available from the Fujitsu Corporation.

Configuration

The ASM or ASM-QFS software configuration process should not be attempted until the Fujitsu LMF software package is installed and working. The following sections describe other aspects of configuration that are specific to the Fujitsu LMF automated library operations.

The mcf File

When creating the mcf file to include one or more Fujitsu LMF automated libraries, the Equipment Identifier field in the mcf file must be the full path name to the parameters file used by the fujitsulmf interface.

Each automated library in the ASM or ASM-QFS environment must have its own identification line in the mcf file. To specify more than one path name in the Equipment Identifier field, put the paths on separate lines.

The Fujitsu LMF Parameters File

The Fujitsu LMF parameters file identifies the drives in the automated library. There must be one parameters file for each automated library. The name of the parameters file must be `/etc/opt/SUNWsamfs/`*eq*, where *eq* is the Equipment Ordinal as defined in the `mcf` file.

The parameters file consists of `lmfdrive` *drivename* = *value* definition lines and comment lines. There must be one `lmfdrive` line for each drive assigned to the client automated library. Comments can appear anywhere on any line, but they must be preceded by a pound sign (#). Characters to the right of the pound sign are ignored.

All *drivename* and *value* information is case sensitive. Table 12. shows the values you can specify for *drivename* and *value*.

Table 12. The *drivename* and *value* Arguments

Argument	Definition
drivename	The name of the drive according to the LMF configuration.
value	The path to the drive. This path must match the Equipment Identifier field of the `mcf` file.

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

Example. The following `mcf` entries define a Fujitsu LMF automated library:

```
#
# Sample mcf file entries for an LMF library
#
/etc/opt/SUNWsamfs/lmf50 50  fj  fj50  -  /etc/opt/SUNWsamfs/
fj50_cat
      /dev/rmt/0cbn      51  fd  fj50  -  /dev/samst/c2t5u0
      /dev/rmt/1cbn      52  fd  fj50  -  /dev/samst/c2t6u0
The following is the parameters file referenced by the
preceding mcf file:
#
# This is file /etc/opt/SUNWsamfs/lmf50
#
# The name "LIB001DRV000" is from the LMF configuration.
#
lmfdrive LIB001DRV000 = /dev/rmt/0cbn   # defines first drive
#
# the name "LIB001DRV001" is from the LMF configuration
#
lmfdrive LIB001DRV001 = /dev/rmt/1cbn  # defines second drive
```

To Build a Catalog

The ASM and ASM-QFS systems automatically build a library catalog for a Fujitsu LMF automated library. However, you must populate the library catalog. There are two ways of doing this.

- Method 1: Use a series of `import`(1M) commands. Note that the cartridges must be physically present in the Fujitsu LMF automated library for the `import`(1M) commands to be successful. If a cartridge is not present, the entry goes into the historian.

  The following series of commands populates the library catalog with entries for three example VSNs:

```
# import -v vsn1 50
# import -v vsn2 50
# import -v vsn3 50
```

- Method 2: Use the `build_cat`(1M) command. This alternate method can be used for large automated libraries with many cartridges. You can create a file containing a list of VSNs and run the `build_cat`(1M) command on that file. This command uses the VSN list to populate the initial catalog. For more information on this, see the `build_cat`(1M) man page.

Note that the slot position of the tape in the Fujitsu LMF automated library has no relationship to the slot number of the VSN in the ASM or ASM-QFS library catalog.

The following example file shows the format of the file to be used by the `build_cat`(1M) command. This example file has a list of VSNs to populate a library catalog. The first column is the ASM or ASM-QFS library catalog slot number, followed by the label, the bar code, and the media type. This example file is as follows:

```
0 VSN186 VSN186 fd
1 VSN187 VSN187 fd
2 VSN188 VSN188 fd
3 VSN189 VSN189 fd
```

The `audit`(1M) command is not supported for Fujitsu LMF automated libraries.

Operations

“Basic Operations” on page 13 describes most of the basic operations that can be performed with an automated library. These operations include labeling cartridges, loading cartridges, and so on. From within a ASM or ASM-QFS environment, most basic operations are the same on most automated libraries. However, some basic operations for the Fujitsu LMF automated libraries differ from typical basic operations. These basic operations differ only in the area of importing and exporting cartridges.

Because you use vendor-supplied utilities to physically add or remove cartridges in the Fujitsu LMF automated library, the ASM and ASM-QFS `import`(1M) and `samexport`(1M) commands and the `libmgr`(1M) import and export menus affect only the library catalog.

You can perform the import and export procedures using `samu`(1M), `robottool`(1M) or `libmgr`(1M). For more information on these tools, see the `samu`(1M), `robottool`(1M), or `libmgr`(1M) man pages. For more information on the `import`(1M) and `samexport`(1M) commands, see the `import`(1M) and `samexport`(1M) man pages.

The import and export processes are explained in the following sections.

To Import a Cartridge

To import a cartridge, perform the following steps.

1. Use Fujitsu commands to physically move the cartridge into the library.
2. Use the ASM and ASM-QFS `import`(1M) command to update the library catalog.

   The syntax for the `import`(1M) command is as follows:

```
import -v volser eq
```

where:

volser	The *volser* to be added. The `fujitsulmf` interface verifies that the LMF interface has the *volser* information before updating the library catalog with the new entry.
eq	The Equipment Ordinal of the device being addressed as defined in the `mcf` file.

To Export a Cartridge

To export a cartridge, perform the following steps.

1. Use the ASM and ASM-QFS `samexport`(1M) command to remove the entry from the library catalog.

   The syntax for the `samexport`(1M) command is one of the following:

```
samexport eq:slot
samexport media_type.vsn
```

where:

eq	The Equipment Ordinal of the device being addressed as defined in the `mcf` file.
slot	The number of a storage slot in an automated library as recognized in the library catalog.
media_type	The media type. For a list of valid media types, see the `mcf`(4) man page.
vsn	The volume serial name assigned to the volume.

The `samexport`(1M) command updates the library catalog as each VSN is exported, and it moves the library catalog entry for each VSN from the ASM or ASM-QFS library catalog to the ASM or ASM-QFS historian.

2. Use Fujitsu commands to physically move the cartridge out of the library.

IBM 3584 UltraScalable Tape Library

The IBM 3584 UltraScalable Tape Library is supported in the ASM and ASM-QFS environments. The following sections describe how to implement cleaning and how to work with this library's partitioning features when this library is used in a ASM or ASM-QFS environment.

Importing

When the ASM or ASM-QFS software is started, cartridges that are in the mailbox are not automatically imported.

Cleaning

In order to use this library in a ASM or ASM-QFS environment, you need to disable automatic cleaning and enable hosted cleaning. This process is described in the *IBM 3584 UltraScalable Tape Library Planning and Operator Guide*, IBM publication GA32-0408-01. This is also described in the `ibm3584`(7) man page, which is distributed by Sun Microsystems.

Partitioning

This library accommodates several tape drives. If you are using multiple drives, it is possible to divide this one physical library into two, three, or four logical libraries. If you have divided your library into two or more logical libraries, be sure that these logical libraries are operating properly before you add the IBM 3584 library to the ASM or ASM-QFS environment.

When a cartridge is exported from a partitioned library, only the logical library from which it was exported can access that drawer slot. If the cartridge is removed and reinserted manually, it is accessible to any and all logical partitions. The following steps describe the act of removal as used in this situation:

1. Open the door
2. Remove the cartridge(s)
3. Close the door
4. Wait for the door to lock and then unlock
5. Open the door
6. Replace the cartridge(s)
7. Close the door

For more information on using this library as a logically partitioned library in a ASM or ASM-QFS environment, see your IBM documentation or see the `ibm3584`(7) man page.

IBM 3494 Libraries

The IBM 3494 automated tape library operates in ASM and ASM-QFS environments with the assistance of the IBM `lmcpd` package. This software must be obtained from IBM, and it must be installed and running before you install the `SUNWsamfs` package from StorageTek.

In addition to installing the preceding software package, you must configure a `/etc/ibmatl.conf` file and a parameters file for the library.

For more information on configuring the IBM 3494 library to work in a ASM or ASM-QFS environment, see the `ibm3494`(7) man page.

Sony Direct-Attached B9 and B35 Automated Libraries

The Sony B9 and B35 series automated libraries contain DTF drives. These stacker libraries optionally use barcode readers.

Note: The information in this section applies only to Sony direct-attached B9 and B35 automated libraries. This information does not pertain to the “Sony Direct-Attached 8400 PetaSite Automated Libraries” on page 59 or to the “Sony Network-Attached Automated Libraries” on page 62.

Configuration

The Sony B9 and B35 series automated libraries should not be configured with autoclean or autoload when running in a ASM or ASM-QFS environment. For more information, see “To Clean a Tape Drive” on page 36.

The autoload feature can be used during the initial loading of cartridges as long as the ASM or ASM-QFS systems is not running. Remember to disable autoload when the system is running.

Operations

Operations on the Sony automated libraries vary depending on whether a mailbox is available. The mailbox is used to import and export cartridges from the automated library. Use the utilities described in “Basic Operations” on page 13 to perform importing, exporting, and other tasks.

■ Sony Direct-Attached 8400 PetaSite Automated Libraries

The Sony 8400 PetaSite Series automated library is different from other Sony models because it has an eight-slot import and export mailbox (slots 400–407). For this reason, the import and export operations are more straightforward for this system. This automated library uses a barcode reader.

Because the mailbox slots can be used as storage slots, the ASM and ASM-QFS library catalogs keep track of the mailbox slots.

Note: The information in this section applies only to Sony direct-attached 8400 PetaSite automated libraries. This information does not pertain to the “Sony Direct-Attached B9 and B35 Automated Libraries” on page 58 or to the “Sony Network-Attached Automated Libraries” on page 62.

To Import Tapes

To import tapes, follow these steps.

1. Open the door of the automated library by pushing the open/close button on the front panel of the automated library.

2. Load the cartridges into the mailbox slots.

3. Push the open/close button on the front panel of the automated library and manually close the door to the mailbox.

   The automated library checks the mailbox slots for the cartridge barcodes after the door is closed. If there is a problem with the barcodes, both the `in` and `out` lights flash for that slot.

4. Use the `import`(1M) command to enable the ASM or ASM-QFS system to recognize the imported cartridges.

   The format of this command is as follows:

```
import eq
```

where:

eq	The Equipment Ordinal of the device being addressed as defined in the `mcf` file.

You can also perform this step by using `libmgr` or `robottool`. For more information on these tools, see the `libmgr`(1M) or `robottool`(1M) man pages.

To Export Tapes

The procedure for exporting tape cartridges differs depending on whether or not you are using the mailbox slots as storage slots.

To Export a Tape Without Using the Mailbox Slots As Storage Slots

Use the following procedure to export a cartridge when you are not using the mailbox slots as storage slots.

1. Issue the `move`(1M) command to move the cartridge to a mailbox slot (slot 400-407).

   This command has the following format:

```
move source_slot destination_slot eq
```

where:

source_slot	The slot number of the slot in which the cartridge currently resides.
destination_slot	The slot number of the slot into which the cartridge should be moved.
eq	The Equipment Ordinal of the device being addressed as defined in the `mcf` file.

2. Push the open/close button on the front panel of the automated library.

The door opens.

3. Remove the cartridge from the mailbox slot.
4. Push the open/close button on the front panel of the automated library and manually close the door to the mailbox.
5. Issue the `samexport`(1M) command to enable the ASM or ASM-QFS system to recognize the exported cartridge.

   This command has the following format:

```
samexport eq
```

where:

eq	The Equipment Ordinal of the device being addressed as defined in the `mcf` file.

You can also perform this using `libmgr`(1M) or `robottool`(1M). For more information on these tools, see the `libmgr`(1M) or `robottool`(1M) man pages.

To Export a Tape Using Mailbox Slots as Storage Slots

Use the following procedure to export a cartridge when you are using the mailbox slots as storage slots and the cartridge you want to export is in one of the mailbox slots.

1. Push the open/close button on the front panel of the automated library.

   The door opens.

2. Remove the cartridge from the mailbox slot.
3. Push the open/close button on the front panel of the automated library and manually close the mailbox door.
4. Issue the `samexport`(1M) command to enable the ASM or ASM-QFS system to recognize the exported cartridge.

   This command has the following format:

```
samexport eq
```

where:

eq	The Equipment Ordinal of the device being addressed as defined in the `mcf` file.

You can also perform this step using `libmgr`(1M) or `robottool`(1M). For more information on these tools, see the `libmgr`(1M) or `robottool`(1M) man pages.

How to Move a Cartridge to a Different Slot

To move a cartridge to a different slot, follow these steps:

1. Make sure that the source slot is occupied and that the destination slot is empty.
2. Issue the `move`(1M) command.

   This command has the following format:

```
move eq:source_slot destination_slot
```

where:

eq	The Equipment Ordinal of the device being addressed as defined in the `mcf` file.
source_slot	The slot number of the slot in which the cartridge currently resides.
destination_slot	The slot number of the slot into which the cartridge should be moved.

You can also perform this step using `libmgr`(1M) or `robottool`(1M). For more information on these tools, see the `libmgr`(1M) or `robottool`(1M) man pages.

Sony Network-Attached Automated Libraries

A Sony network-attached automated library operates within the ASM and ASM-QFS environments through the DZC-8000S Application Interface Library package. This software package provides the application programmer interface (API) to the PetaSite Controller (PSC). For more information on the DZC-8000S interface, see the *Sony PetaSite Application Interface Library DZC-8000S*, which is available from Sony.

Note: The information in this section applies only to Sony automated libraries that are network-attached through a Sony DZC-8000S interface. This information does not pertain to direct-attached Sony automated libraries. For more information on direct-attached Sony libraries, see “Sony Direct-Attached B9 and B35 Automated Libraries” on page 58 or “Sony Direct-Attached 8400 PetaSite Automated Libraries” on page 59.

Configuration

The following sections describe other aspects of configuration that are specific to Sony automated library operations.

The mcf File

When you create the `mcf` file to include one or more Sony network-attached automated libraries that use the DZC-8000S interface, the `Equipment Identifier` field in the `mcf` file must be the full path name to the parameters file used by the DZC-8000S interface.

Sony Parameters File

The Sony parameters file consists of a list of *keyword* = *value* parameter lines. The various *keyword* values identify the Sony automated libraries, the drives associated with the libraries, and the host name. All *keyword* and *value* entries are case sensitive, so they must be entered exactly as used in the configuration file and in the ASM or ASM-QFS `mcf` file. The following types of *keyword* = *value* parameters must appear in the Sony parameters file:

- `userid` = *user_id*. For *user_id* specify a number in the range 0 ≤ *user_id* ≤ 65535. The `userid` parameter identifies the user during initialization of the PetaSite automated library functions. This is a required parameter.
- `server` = *server_id*. For *server_id* specify the host name of the server running the PSC server code. This is a required parameter.
- `sonydrive` *drive_id* = *path* [shared]. This is a required parameter; there must be one `sonydrive` line for every drive defined in the `mcf` file.

  For *drive_id*, specify the drive bin number as configured in the PSC configuration file.

  For *path*, specify the path to the drive as entered in the `Equipment Identifier` field of the ASM or ASM-QFS `mcf` file.

  The `shared` keyword is optional. This library can be configured to share its media drives with two or more ASM or ASM-QFS processes from two or more hosts. For more information on implementing shared drives, see “Shared Drives” on page 18 or see the `sony`(7) man page.

Comments can appear anywhere on any line, but they must be preceded by a pound sign (#). Characters to the right of the pound sign are ignored.

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

Example. This example shows a ASM `mcf` file and a Sony parameters file. The `mcf` file is as follows:

```
#
# Sample mcf file entries for a Sony network-attached library
#
/etc/opt/SUNWsamfs/sonyfile  100   pe   psc  on
/dev/rmt/1cbn                101   so   psc  on
/dev/rmt/2cbn                102   so   psc  on
```

The following parameters file, `/etc/opt/SUNWsamfs/sonyfile`, is the file referenced by the preceding `mcf` file:

```
#
# This is file: /etc/opt/SUNWsamfs/sonyfile
#
# The userid identifies the user during initialization of
# the PetaSite library functions
#
userid = 65533
#
# europa is the hostname for the server running
# the DZC-8000S server code.
#
server = europa
#
# The bin numbers 1001 and 1002 are from the PSC
# configuration file.
#
sonydrive 1001 = /dev/rmt/1cbn
sonydrive 1002 = /dev/rmt/2cbn shared
```

To Build a Catalog

Like a direct-attached automated library, the ASM and ASM-QFS systems automatically build a library catalog for a Sony automated library. However, you must populate the library catalog. There are two ways of doing this.

- Method 1: Use a series of `import`(1M) commands. Note that the cartridges must be physically present in the Sony automated library for the `import`(1M) commands to be successful. If a cartridge is not present, the entry goes into the historian.

  If a VSN name contains one or more space characters, the VSN name must be enclosed in quotation marks (“ “).

The following series of commands populates the library catalog with entries for three example VSNs:

```
# import -v “SEG 99001” 50
# import -v vsn2 50
# import -v vsn3 50
```

- Method 2: Use the build_cat(1M) command. This alternate method can be used for large automated libraries with many cartridges. You can create a file containing a list of volume serial names (VSNs) and run the build_cat(1M) command on that file. This command uses the VSN list to populate the initial catalog. For more information on this, see the build_cat(1M) man page.

  Note that the slot position of the tape in the Sony automated library has no relationship to the slot number of the VSN in the library catalog.

  If a VSN name contains one or more space characters, the VSN name must be enclosed in quotation marks (“ “).

Example. The following example file shows the format of the file to be used by the build_cat(1M) command. This example file has a list of VSNs to populate a library catalog. The first column is the ASM or ASM-QFS library catalog slot number, followed by the label, the bar code, and the media type. This example file is as follows:

```
0 A00001 “2000 B00001” so
1 A00002 B00002 so
2 TEST01 TEST01 so
3 TEST02 TEST02 so
```

The build_cat(1M) command is not supported for Sony network-attached automated libraries.

Operations

“Basic Operations” on page 13 describes most of the basic operations that can be performed with an automated library. These operations include labeling cartridges, loading cartridges, and so on. From within the ASM or ASM-QFS environment, most basic operations are the same on most automated libraries. However, some basic operations for Sony automated libraries differ from typical basic operations. These basic operations differ only in the area of importing and exporting cartridges.

Because you use vendor-supplied utilities to physically add and remove cartridges in the Sony automated library, the ASM import(1M) and samexport(1M) commands and the ASM libmgr(1M) import and export menus affect only the library catalog.

You can perfom the import and export procedures using `samu`(1M), `robottool`(1M), or `libmgr`(1M). For more information on these tools, see the `samu`(1M), `robottool`(1M), or `libmgr`(1M) man pages. For more information on the `import`(1M) and `samexport`(1M) commands, see the `import`(1M) and `samexport`(1M) man pages.

The import and export processes are described in the following sections.

To Import a Cartridge

To import a cartridge, perform the following steps.

1. Use Sony commands to physically move the cartridge into the library.
2. Use the ASM and ASM-QFS `import`(1M) command to update the library catalog.

   The syntax for the `import`(1M) command is as follows:

```
import -v [“ ] volser [ “ ] eq
```

where:

“ “	Quotation marks. The *volser* must be enclosed in quotation marks if it contains spaces.
volser	The *volser* to be added. The PSC API interface verifies that the Sony automated library has the volser information before updating the library catalog with the new entry. If the cartridge does not physically exist in the library, the entry is placed in the historian catalog.
eq	The Equipment Ordinal of the library being addressed as defined in the `mcf` file.

To Export a Cartridge

To export a cartridge, perform the following steps.

1. Use the ASM and ASM-QFS `samexport`(1M) command to remove the entry from the library catalog.

   The syntax for the `export`(1M) command is one of the following:

```
samexport eq:slot
samexport media_type.vsn
```

where:

eq	The Equipment Ordinal of the device being addressed as defined in the `mcf` file.
slot	The number of a storage slot in an automated library as recognized in the library catalog.
media_type	The media type. For a list of valid media types, see the `mcf`(4) man page.
vsn	The volume serial name assigned to the volume.

The `samexport`(1M) command updates the library catalog as each VSN is exported, and it moves the library catalog entry for each VSN from the library catalog to the historian.

2. Use Sony commands to physically move the cartridge out of the library.

StorageTek ACSLS-Attached Automated Libraries

In many respects, the ASM and ASM-QFS systems interoperate with a StorageTek ACSLS-attached automated library in a manner similar to direct-attached automated libraries. However, a StorageTek ACSLS-attached automated library requires additional steps in the installation and configuration procedure as compared to a direct-attached automated library.

The ACSLS software package supplied by StorageTek controls the automated library. Daemon software controls the StorageTek automated library through the ACSAPI interface.

The following sections describe how to manage StorageTek ACSLS-attached automated libraries. Topics presented include configuration, basic operations, error messages, and other topics.

Configuration

The ASM or ASM-QFS software configuration process should not be attempted until the StorageTek ACSLS software package is installed and working. The following sections describe other aspects of configuration that are unique to StorageTek ACSLS-attached automated libraries.

The mcf File

The `mcf` for StorageTek ACSLS-attached automated libraries differs from direct-attached libraries in the following fields:

- The `Equipment Identifier` field contains the full path name to a parameters file used by the `stk` daemon. This parameters file defines the system characteristics of the StorageTek automated library and its drives within the ASM or ASM-QFS environment. The content of this file is defined in the “StorageTek Parameters File” on page 68.

- The `Equipment Type` field contains a two-character mnemonic for the device type. For ACSLS-attached libraries, you must use the `sk` equipment type. For other Equipment Types, see the `mcf`(4) man page.
- The `Family Set` field contains the names of the drives associated with the automated library.

Figure 5. shows entries in an `mcf` file that define a StorageTek automated library with two associated 9840 drives.

Figure 5. StorageTek Library Entries in an `mcf` File

```
# Equipment               Eq   Eq  Family   Dev  Additional
# Identifier              Ord  Ty  Set      St   Parameters
#
/etc/opt/SUNWsamfs/stk50 50   sk  sk50     on
/dev/rmt/0cbn             51   sg  sk50     on
/dev/rmt/1cbn             52   sg  sk50     on
```

StorageTek Parameters File

During configuration, you must create a parameters file for each ACSLS-attached StorageTek automated library. Each line of the parameters file must begin with a keyword or a comment. The keywords to use are as follows:

- `access =` *user_id*

  Specifies the user identification. For *user_id*, enter the *user_id* used by StorageTek for access control. This is an optional entry to the parameters file. If the `access =` parameter is not supplied, the access control string is a null string. This indicates that there is no *user_id*.

- `hostname` = *host_name*

  Specifies the host name of the server. For *host_name*, enter the host name of the server running the StorageTek ACSLS interface. See the `README` file distributed with the release for information on the value to include for *host_name*.

- `portnum =` *port_number*

  Specifies the port number used for communication between ACSLS and the ASM or ASM-QFS software. See the `README` file distributed with the release for information on the value to include for *port_number*.

- `capacity =` (*index* = *value*[, *index* = *value*] . . .)

  Sets the capacity of the cartridges supported by StorageTek. The *index* = *value* pairs must be separated by commas and enclosed in parentheses.

For *index*, specify the index of the `media_type` file supplied by StorageTek and located in the following ACSLS directory:

`/export/home/ACSSS/data/internal/mixed_media/media_types.dat`

For *value*, enter the capacity of the cartridge type in units of 1024 bytes. The ASM and ASM-QFS systems have defaults for *index* that were current at the time of the ASM and ASM-QFS 4.0 releases. In general, it is necessary only to supply a capacity entry for an index of new cartridge types or to override the capacity supported by StorageTek.

Table 13. shows the defaults.

Table 13. Capacity Defaults

index	**Type**	**Capacity**
0	3480	210 MB (215040)
1	3490E	800 MB (819200)
2	DD3A	10 GB (10485760)
3	DD3B	25 GB (26214400)
4	DD3C	50 GB (52428800)
5	DD3D	0 (DD3 cleaning tape)
6	DLTIII	10 GB (10485760)
7	DLTIV	20 GB (20971520)
8	DLTIIIXT	15 GB (15728640)
9	STK1R (9840)	20 GB (20971520)
10	STK1U	0 (STK1R cleaning tape)
11	EECART	1.6 GB (16777216)
12	JCART	0 GB (foreign label)
13	STK2P (T9940A)	60 GB (62914560)
14	STK2W	0 GB (T9940A cleaning tape)
15	KLABEL	0 GB (unsupported)
16	LTO-100G	100 GB (104857600)
17	LTO-50G	50 GB (52428800)
18	LTO-35G	35 GB (36700160)
19	LTO-10G	10 GB (10485760)
20	LTO-CLN2	0 GB (cleaning tape)
21	LTO-CLN3	0 GB (cleaning tape)

Table 13. Capacity Defaults (Continued)

index	**Type**	**Capacity**
22	LTO-CLN1	0 GB (cleaning tape)
23	SDLT	110 GB (115343360)

- `device_path_name =` (`acs =` *value*, `lsm =` *value*, `panel =` *value*, `drive =` *value*) [`shared`]

  Specifies the path to the device on the client. There must be one `device_path_name` = entry for each drive attached to this client. The description of the drive within the StorageTek automated library follows the `device_path_name` = keyword. This description starts with an open parenthesis followed by four *keyword = value* pairs and a closed parenthesis.

  The *keyword = value* pairs can be separated by a comma (as shown), a colon, or a space. Use the information supplied by the ACSLS query drive command to configure the `device_path_name`. Table 14. shows the *value* specifications.

Table 14. The *value* Specifications

Value	**Content**
`acs`	ACS number for the drive as configured in the StorageTek library
`lsm`	LSM number for the drive as configured in the StorageTek library
`panel`	PANEL number for the drive as configured in the StorageTek library
`drive`	DRIVE number for the drive as configured in the StorageTek library

The `shared` keyword can follow the specification for the device path name. This specifies that the drive can be shared between two or more ASM or ASM-QFS processes from two or more hosts. For more information on implementing shared drives, see “Shared Drives” on page 18 or see the `stk`(7) man page.

The following is an example of a parameters file for a StorageTek automated library:

```
#
#  This is file: /etc/opt/SUNWsamfs/stk50
#
hostname = baggins
portnum = 50014
access = some_user  # No white space allowed in user_id
capacity = ( 7 = 20971520, 9 = 20971520 )
/dev/rmt/0cbn = (acs=0, lsm=1, panel=0, drive=1) shared
/dev/rmt/1cbn = (acs=0, lsm=1, panel=0, drive=2)
```

The ssi.sh Script

The sam-stkd daemon uses ssi.sh to ensure that a copy of the SSI daemon, ssi_so, is running. If ssi_so exits, the daemon starts another. If your site has its own version of ssi.sh, this script should be modified to wait for a SIGTERM signal and then exit. SIGTERM is the signal sent by the daemon to stop the process.

An example ssi.sh script can be found in /opt/SUNWsamfs/examples/ssi.sh. The ssi.sh script is automatically copied to /etc/opt/SUNWsamfs/ssi.sh during installation if one does not already exist.

To Build a Catalog

The ASM and ASM-QFS systems automatically build a library catalog for a StorageTek automated library. However, you must populate the library catalog. There are two ways of doing this.

- Method 1: Use a series of import(1M) commands. Note that the cartridges must be physically present in the StorageTek ACSLS-attached library for the import(1M) commands to be successful. If a cartridge is not present, the entry goes into the historian.

  The following series of commands populates the library catalog with entries for three example VSNs:

```
# import -v vsn1 50
# import -v vsn2 50
# import -v vsn3 50
```

- Method 2: Use the build_cat(1M) command. This alternate method can be used for large automated libraries with many cartridges. You can create a file containing a list of volume serial names (VSNs) and run the build_cat(1M) command on that file. This command uses the VSN list to populate the initial catalog. For more information on this, see the build_cat(1M) man page.

Note that the slot position of the tape in the ACSLS-attached StorageTek automated library has no relationship to the slot number of the VSN in the library catalog.

The following example file shows the format of the file to be used by the `build_cat`(1M) command. This example file has a list of VSNs to populate a library catalog. The first column is the library catalog slot number, followed by the label, the bar code, and the media type. This example file is as follows:

```
0 DLT186 DLT186 lt
1 DLT187 DLT187 lt
2 DLT188 DLT188 lt
3 DLT189 DLT189 lt
```

The `audit`(1M) command is not supported for ACSLS-attached automated libraries.

Common Problems and Error Messages

The following examples show common problems and the messages that the system generates when they are encountered.

Example 1. The following messages are generated when there are syntax errors in the StorageTek parameters file. Check your StorageTek parameters file for syntax errors and remember that each line must begin with a keyword or a comment. For more information on the StorageTek parameters file, see the `stk`(7) man page.

```
May 23 09:26:13 baggins stk-50[3854]: initialize: Syntax
error in stk configuration file line 4.
May 23 09:26:13 baggins stk-50[3854]: initialize: Syntax
error in stk configuration file line 5.
```

Example 2. You receive two sets of error messages. The following is the first set:

```
May 23 09:29:48 baggins stk-50[3854]: main: Waiting for 2
drive(s) to initialize
May 23 09:29:59 baggins stk-50[3854]: main: Waiting for 2
drive(s) to initialize
May 23 09:30:39 baggins stk-50[3854]: main: Waiting for 2
drive(s) to initialize
```

The following is the second set:

```
May 23 09:31:19 baggins stk-50[3854]: main: 2 drive(s) did
not initialize.
```

The samu(1M) utility's :r display is as follows:

```
ty   eq  status      act  use  state    vsn
sg   51  ---------p    0   0%  off
        drive set off due to ACS reported state
sg   52  ---------p    0   0%  off
        drive set off due to ACS reported state
lt   61  ---------p    0   0%  off
        drive set off due to ACS reported state
tp   62  ----------    0   0%  off
        empty
```

Drives that are hung in an initializing state or that do not initialize usually indicate a configuration error. Verify that ACSLS is up and running. Verify the host name. Determine whether you can you ping the host name using the ping(1M) command.

Check portnum in the StorageTek parameters file. In ACSLS 5.3, for example, the default port number, 50004, is used for a different application. Try a higher port number, such as 50014.

Example 3. In this example, the import(1M) command was used to import a VSN to the library catalog, but the VSN is not in the StorageTek automated library. The cartridge must be present in the ACSLS-managed automated library before the import(1M) command can be successful. The following messages are generated:

```
May 20 15:09:33 baggins stk-50[6117]: view_media
returned:STATUS_VOLUME_NOT_IN_LIBRARY
May 20 15:09:33 baggins stk-50[6117]: add_to_cat_req:
view_media: failed:STATUS_VOLUME_NOT_IN_LIBRARY. A
```

Operations

"Basic Operations" on page 13 describes most of the basic operations that can be performed with an automated library. These operations include labeling cartridges, loading cartridges, and so on. From within the ASM or ASM-QFS environment, most basic operations are the same on most automated libraries. The following sections describe basic operations for the StorageTek ACSLS-attached automated libraries that differ only in the area of importing and exporting cartridges.

A *mailbox* is an area used for putting cartridges into and removing cartridges from the automated library. Some StorageTek automated libraries import and export only one cartridge at a time. Examples of StorageTek automated libraries with a mailbox that are supported within the ASM and ASM-QFS environments include the StorageTek 9714 and the StorageTek 9710. The StorageTek 9730 uses a mailslot. In StorageTek documentation, the mailbox and mailbox slot are often referred to as the *CAP*.

When importing and exporting cartridges from an ACSLS-attached automated library, ASM and ASM-QFS commands affect only the library catalog. The cartridges are not physically inserted into or removed from the automated library by the ASM and ASM-QFS commands. You must use the ACSLS commands to physically move the cartridges. It is your responsibility to keep the ACSLS inventory and the ASM or ASM-QFS catalog in agreement.

You can also perform the import and export procedures using `samu`(1M), `robottool`(1M) or `libmgr`(1M). For more information on these tools, see the `samu`(1M), `robottool`(1M), or `libmgr`(1M) man pages.

To Import Tapes

- To import tape cartridges, use the `import`(1M) command.

This command has the following format:

```
import -v vsn eq
```

where:

vsn	The volume serial name assigned to the volume.
eq	The Equipment Ordinal of the device being addressed as defined in the `mcf` file.

The `import`(1M) command causes the new VSN to appear in the library catalog. If the VSN was in the historian, the ASM or ASM-QFS software moves the VSN information from the historian to the library catalog.

To Export Tapes Using a Mailbox

You can export tape cartridges by slot or by VSN.

- To export tape cartridges, use the `samexport`(1M) command.

This command has the following formats:

```
samexport eq:slot
samexport media_type.vsn
```

where:

eq	The Equipment Ordinal of the device being addressed as defined in the `mcf` file.
slot	The number of a storage slot in an automated library as recognized in the library catalog.
media_type	The media type. For a list of valid media types, see the `mcf`(4) man page.
vsn	The volume serial name assigned to the volume.

The `samexport`(1M) command updates the library catalog as each VSN is exported, and it moves the library catalog entry for each VSN from the library catalog to the historian.

Archiving

4

Archiving is the process of copying a file from a ASM or ASM-QFS file system to a volume that resides on a removable media cartridge or on a disk partition of another file system. Throughout this chapter, the term *archive media* is used to refer to the various cartridges or disk slices to which archive volumes are written. The ASM and ASM-QFS archiving capabilities include many features, such as those that you can use to specify that files be archived immediately, to specify that files never be archived, and to perform other tasks.

This chapter describes the archiver's theory of operations, provides general guidelines for developing archive policies for your site, and explains how to implement policies by creating an `archiver.cmd` file.

The following topics are presented:

Archiver – Theory of Operations

The archiver automatically archives ASM and ASM-QFS files to archive media. Operator intervention is not required to archive and stage the files. Files are archived to a volume on the archive media, and each volume is identified by a unique identifier called a *volume serial name* (VSN). Archive media can contain one or more volumes. To identify an individual volume, the media type and VSN must be specified.

The archiver starts automatically when a ASM or ASM-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`

The `archiver.cmd` file does not need to be present for archiving to occur. In the absence of this file, the ASM and ASM-QFS environments use the following defaults:

- All files are archived to available volumes.
- The *archive age* for all files is 4 minutes. The archive age is the amount of time since a file's last modification.
- The *archive interval* is 10 minutes. The archive interval is the amount of that elapses between complete archiving processes.

The following sections describe the concept of an archive set and explain the operations performed during the archiving process.

Archive Sets

An *archive set* identifies a group of files to be archived. Archive sets can be defined across any group of file systems. Files in an archive set share common criteria that pertain to the size, ownership, group, or directory location. The archive sets control the destination of the archive copy, how long to keep the copy archived, and how long to wait before archiving the data. All files in an archive set are copied to the volumes associated with that archive set. A file in the file system can be a member of one and only one archive set.

As files are created and modified, the archiver copies them to archive media. Archive files are compatible with the standard UNIX `tar`(1) format. This ensures data compatibility with the Solaris operating environment (OE) and other UNIX systems. This format includes the file access data (inode) and the path to the file. If a complete loss of your ASM or ASM-QFS environment occurs, the `tar`(1) format allows file recovery using standard UNIX tools and commands. The archiving process also copies the data necessary for ASM and ASM-QFS file system operations. This data consists of directories, symbolic links, the index of segmented files, and archive media information.

In the remainder of this section, the term *files* refers to both file data and metadata. The terms *file data* and *metadata* are used only when a distinction is required. The term *file system* refers to a mounted ASM or ASM-QFS file system.

Archive set names are determined by the administrator and are virtually unlimited with the following exceptions:

- There are two reserved archive set names: `no_archive` and `allsets`.

  The `no_archive` archive set is defined by default. Files selected to be in this archive set are never archived. Files in a temporary directory, such as `/sam1/tmp`, for example, might be included in the `no_archive` archive set.

  The `allsets` archive set is used to define parameters that apply to all archive sets.

- Archive sets named for each ASM or ASM-QFS file system are reserved for control structure information. ASM and ASM-QFS file systems provide a default archive set for each file system. For each file system, both the metadata and data files are archived. The file system archive set encompasses the directory and link information and any files that are not included in another archive set. The default archive sets are given the names of their associated file systems and cannot be changed. For example, `samfs1` would be the archive set name for a file system configured and named as `samfs1`.
- Archive set names are limited to 29 characters. The characters are limited to the 26 uppercase and lowercase letters, the digits 0 through 9, and the underscore character (_).

Archiving Operations

By default, the archiver makes one copy of each archive set, but you can request up to four archive copies for each archive set. An archive set and a copy number become a synonym for a collection of volumes. The archive copies provide duplication of files on separate volumes.

To ensure that files are complete before archiving, the archiver waits a specified period of time after the file is modified before archiving it. As mentioned previously, this period of time is called the *archive age*.

The data in a file must be modified before the file is considered to be a candidate for archiving or rearchiving. A file is not archived if it is only accessed. For example, issuing a `touch`(1) or an `mv`(1) command on a file does not cause it to be archived or rearchived. Issuing an `mv`(1) command alters the file name but not the file data, and this can have ramifications in a disaster recovery situation if you are restoring from `tar`(1) files. For more information on disaster recovery, see the *ASM, ASM-QFS, and ASM/QFS-Standalone Disaster Recovery Guide*.

Files are selected for archiving based on their archive age. The archive age can be defined for each archive copy.

Users can change the default time references on their files to values far in the past or future by using the `touch`(1) command. This can cause unexpected archiving results, however. To avoid such problems, the archiver adjusts the references so that they are always in the following range:

creation_time < *time_ref* < *time_now*

The following sections describe the steps taken by the archiver from the initial file scan to the file copy process.

Step 1: Scanning for Files to Archive

There is a separate `sam-arfind` process for each mounted file system. The `sam-arfind` process scans each file system periodically to determine which files need archiving. The first scan that sam-arfind performs is a directory scan. During this scan, sam-arfind descends recursively through the directory tree. Each file is examined, and the file status flag archdone is set if the file does not need archiving. A file might not need archiving, for example, if all of its archive copies have already been made or if it is a file that is included in the no_archive archive set.

During successive scans, the .inodes file is scanned. Only those inodes with the archdone flag not set are examined. A number of actions can change a file's archive status. Among these actions are the following: a file being rearchived, a file being unarchived, a file being changed, and so on. These actions clear the archdone flag.

The sam-arfind process determines the archive set to which the file belongs by using the file properties descriptions. The characteristics used for determining a file's archive set include the directory path portion of the file's name (and, optionally, the complete file name using a regular expression), the user name of the file's owner, the group name of the file's owner, a minimum file size, and a maximum file size.

If the archive age of the file for one or more copies has been met or exceeded, sam-arfind adds the file to the archive request for the archive set. The archive request is the collection of files that all belong to the same archive set. The archive request is a file that resides in the following directory:

```
/var/opt/SUNWsamfs/archiver/file_sys/ArchReq
```

These are binary files, and you can display them by using the showqueue(1M) command.

The archive request is sometimes referred to as an *ArchReq*.

If a file is offline, the sam-arfind process selects the volumes to be used as the source for the archive copy. If the file copy is being rearchived, the sam-arfind process selects the volume containing the archive copy that is being rearchived.

If a file is segmented, only those segments that have changed are selected for archival. The index of a segmented file contains no user data, so it is treated as a member of the file system archive set and is archived separately.

The archive priority is computed from file property characteristics and from file property multipliers associated with the archive set. Essentially, the computation is as follows:

archive_priority = the sum of (*file_property_value* * *property_multiplier*)

Most *file_property_value* numbers are 1 or 0, as the property is TRUE or FALSE. For instance, the value of the property copy 1 is 1 if archive copy 1 is being made. The values of copy 2, copy 3, and copy 4 are, therefore, 0.

Others, such as archive age and file size, can have values other than 0 or 1.

The *property_multiplier* values are determined from the `-priority` parameters for the archive set. Various aspects of a file, such as age or size, can be given values so that your site can alter the archive request's priority. For more information on the `-priority` parameter, see the `archiver.cmd`(4) man page.

The *archive_priority* and the property multipliers are floating-point numbers. The default value for all property multipliers is 0.0. The archive request is set to the highest file priority in the archive request.

When the file system scanning has been completed, the `sam-arfind` process sends each archive request to the archiver daemon, `sam-archiverd`, to be scheduled for file copying to archive media. The `sam-arfind` process then sleeps for the interval duration. At the end of the interval, the `sam-arfind` process resumes scanning.

If files are found for archiving, and a previous archive request for the archive set has not been completed, the `sam-arfind` process skips the file just found.

Thus, overlapping of file system scanning and file copying take place.

Step 2: Composing Archive Requests

When archive requests are received by the `sam-archiverd` daemon, they are *composed*. This step describes the composition process.

All the files in an archive request might not be archived at one time. This can be caused by the capacity of the archive media or by the controls specified in the archiver command file. *Composing* is the process of selecting the files to be archived from the archive request at one time. When the archive copy operation has been completed for an archive request, the archive request is recomposed if files remain to be archived.

The `sam-archiverd` daemon orders the files in the archive requests according to certain default and site-specific criteria. The default operation is to archive all the files in an archive request to the same archive volumes in the order that they were found during the file system scan. The site-specific criteria allow you to control the order in which files are archived and how they can be distributed on volumes. These criteria are called *archive set parameters*, and the order in which they are evaluated is as follows: `-reserve`, `-join`, `-sort`, and `-drives`. For more information on these parameters, see the `archiver.cmd`(4) man page.

If the archive request belongs to an archive set that has `-reserve` *owner* specified, the `sam-archiverd` daemon orders the files in the archive request according to the file's directory path, user name, or group name. This action is controlled by the `-reserve` parameter for the archive set. The files belonging to the first *owner* are selected for archiving. The remaining files are archived later.

If the archive request belongs to an archive set that has the -join *method* specified, the sam-archiverd daemon groups the files together according to the -join *method* specified. If a -sort *method* is also specified, then the sam-archiverd daemon sorts the files within each group according to the -sort *method*. The archive request is joined and sorted.

Each group of joined files is treated as if it were a single file for the remainder of the composing and scheduling processes.

If the archive request belongs to an archive set that has the -sort *method* specified, the sam-archiverd daemon sorts the files according to the sort method specified on the -sort parameter. Depending on the sort method, the sam-archiverd daemon tends to keep files together based on the sort method, age, size, or directory location. The sam-archiverd daemon marks the archive request as sorted. By default, the archive requests are not sorted, so the files are archived in the order in which they are encountered during the file system scan.

The sam-archiverd daemon determines whether the files are online or offline. If both online and offline files are in the archive request, the online files are selected for archiving first.

If the archive request was not required to be joined or sorted by a sort method, the offline files are ordered by the volume upon which the archive copies reside. This ensures that all files (within each archive set) on the same volume are staged at the same time in the order in which they were stored on the media. When more than one archive copy of an offline file is being made, the offline file is not released until all required copies are made. All the files to be staged from the same volume as the first file are selected for archiving.

Note that using the -sort or -join parameters can have a negative effect on performance when archiving offline files. This is due to the possiblity that the order of the files to be archived does not match the order of the volumes needed for the offline files. It is recommended that you use the -join or -sort parameters only for the first archive copy to be made. Other copies will most likely maintain the order of the first copy if enough archive media is available when the copies are started.

The archive requests are entered in the sam-archiverd daemon's scheduling queue.

Step 3: Scheduling Archive Requests

The scheduler in the sam-archiverd daemon executes on demand when the following conditions exist:

- An archive request is entered in the scheduling queue.
- The archiving for an archive request has been completed.
- A change in media status is received from the catalog server.

- A message is received that changes the state of the archiver.

The archive requests in the scheduling queue are ordered by priority. Each time the scheduler executes, all archive requests are examined to determine if they can be assigned to a `sam-arcopy` process to have the files copied to archive media.

There must be drives available to use for making file copies. There must be volumes available that can be used by the archive set and have sufficient space to hold the files in the archive request.

Drives

If the archive set has the `-drives` parameter specified, the `sam-archiverd` daemon divides the selected files in the archive request among multiple drives. If the number of drives available at this time is less than that specified by the `-drives` parameter, the smaller number is used.

If the total size of files in the archive request is less than the `-drivemin` value, only one drive is used. The `-drivemin` value is either the value specified by the `-drivemin` parameter or it is the `archmax` value.

The `archmax` value is specified by the `-archmax` parameter or the value defined for the media. For more information on the `-archmax` parameter and the `archmax=` directive, see the `archiver.cmd`(4) man page.

If the total size of files in the archive request is more than the the `-drivemin` value, then *drive_count* = *total_size* / *drivemin* computed. If *drive_count* is less than the number of drives computed, the *drive_count* becomes the number of drives to use.

Volumes

There must be a volume, or volumes, with enough space to hold at least some of the files in the archive request. The volume that has most recently been used for the archive set is used if there is enough space. Also, the volume must not be in use by the archiver.

If a volume usable for the archive set is presently busy, another is selected. This is true unless the `-fillvsns` parameter is specified. In this case, the archive request is not schedulable.

If an archive request is too big for one volume, the files that can fit on the volume are selected to be archived to the volume. If the archive request contains files that are too big to fit on one volume, and volume overflow for the archive request is not selected, the files cannot be archived. An appropriate message for this condition is sent to the log.

You can specify volume overflow for the archive set (by using the `-ovflmin` parameter) or for the media (by using the `ovflmin=` directive). For more information on the `-ovflmin` parameter and the `ovflmin=` directive, see the `archiver.cmd`(4) man page. This specification, `ovflmin`, determines the minimum size for files to overflow media. An `ovflmin` specified for the archive set takes precedence over a media-defined `ovflmin`. If the size of the files is less than `ovflmin`, the files cannot be archived. An appropriate message for this condition is sent to the log.

If the size of the files is more than `ovflmin`, the additional volumes are assigned as required. The additional volumes are selected in order of decreasing size in order to minimize the number of volumes required for the file.

If no usable volumes can be found for the archive request, the archive request waits.

Certain properties, such as whether or not the file is online or offline, are used in conjunction with the archive priority (computed in Step 1) when determining the scheduling priority for a particular archive request. For more information on customizing the property multiplier, see the `-priority` parameters described on the `archiver.cmd`(4) man page.

For each archive request, the `sam-archiverd` daemon computes the scheduling priority by adding the archive priority to multipliers associated with various system resource properties. These properties are associated with the number of seconds that the archive request has been queued, whether or not the first volume to be used in the archiving process is loaded into a drive, and so on.

Using the adjusted priorities, the `sam-archiverd` daemon assigns each ready archive request to be copied.

Step 4: Archiving the Files in an Archive Request

When an archive request is ready to be archived, the `sam-archiverd` daemon steps through each archive request to mark the archive file (tarball) boundaries so that each archive file's size is less than the `-archmax` *target_size* specification. If a single file is larger than *target_size*, it becomes the only file in an archive file.

For each archive request and each drive to be used, the `sam-archiverd` daemon assigns the archive request to a `sam-arcopy` process to copy the files to the archive media. If a single file is larger than *target_size*, it becomes the only file in an archive file. The archive information is entered into the inode.

If archive logging is enabled, an archive log entry is created.

If the file was staged, the disk space is released. This process continues until all files in the list have been archived.

A variety of errors and file status changes can prevent a file from being successfully copied. This can include read errors from the cache disk and write errors to the volumes. Status changes include modification since selection, file open for write, and file removed.

When the `sam-arcopy` process exits, the `sam-archiverd` daemon examines the archive request. If any files have not been archived, the archive request is recomposed.

Sample Default Output

The following sample output is from running `archiver -l`:

```
# archiver

Archive media:
default:mo
media:mo archmax:5000000
media:lt archmax:50000000
Archive devices:
device:mo20 drives_available:1 archive_drives:1
device:lt30 drives_available:1 archive_drives:1
Archive file selections:
Filesystem samfs1:
samfs1  Metadata
    copy:1  arch_age:240
big  path:. minsize:512000
    copy:1  arch_age:240
all  path:
    copy:1  arch_age:30
Archive sets:
all
    copy:1 media:mo
big
    copy:1 media:lt
samfs1
    copy:1 media:mo
```

Archiver Daemons

The `sam-archiverd` daemon is responsible for scheduling the archiving activity. The `sam-arfind` process assigns files to be archived to archive sets. The `sam-arcopy` process copies the files to be archived to the selected volumes.

The `sam-archiverd` daemon is started by `ASMd` when ASM or ASM-QFS activity begins. The `sam-archiver` daemon executes the `archiver`(1M) command to read the `archiver.cmd` file and builds the tables necessary to control archiving. It starts a `sam-arfind` process for each mounted file system; likewise, if a file system is unmounted, the associated `sam-arfind` process is stopped. The `sam-archiverd` process then monitors `sam-arfind` and processes signals from an operator or other processes.

Archive Log Files and Event Logging

The sam-arfind and sam-arcopy processes produce a log file that contains information about each archived or automatically unarchived file. The log file is a continuous record of archival action. You can use the log file to locate earlier copies of files for traditional backup purposes.

This file is not produced by default. You can use the logfile= directive in the archiver.cmd file to specify that a log file be created and to specify the name of the log file. You determine the name of this file. For more information on the log file, see the “The archiver.cmd Directives” on page 91 in this chapter and see the archiver.cmd(4) man page.

The archiver logs warnings and informational messages in the log file using the syslog facility and archiver.sharchiver.sh.

The following are sample lines from an archiver log file with definitions for each field:

```
A 2001/03/23 18:42:06 mo 0004A arset0.1 9a089.1329 samfs1
118.51 162514 t0/fdn f 0 56
A 2001/03/23 18:42:10 mo 0004A arset0.1 9aac2.1 samfs1 189.53
1515016 t0/fae f 0 56
A 2001/03/23 18:42:10 mo 0004A arset0.1 9aac2.b92 samfs1
125.53 867101 t0/fai f 0 56
A 2001/03/23 19:13:09 lt SLOT22 arset0.2 798.1 samfs1
71531.14 1841087 t0/fhh f 0 51
A 2001/03/23 19:13:10 lt SLOT22 arset0.2 798.e0e samfs1
71532.12 543390 t0/fhg f 0 51
```

Reading left to right, the fields in the previous listing have the content shown in Table 15..

Table 15. Archiver Log File Fields

Field	Content
1	Archive activity, as follows: • A for archived. • R for rearchived. • U for unarchived.
2	Date of archive action in *yyyy/mm/dd* format.
3	Time of archive activity in *hh:mm:ss* format.
4	Archive media type. For information on media types, see the mcf(4) man page.
5	VSN.
6	Archive set and copy number.

Table 15. Archiver Log File Fields (Continued)

Field	Content
7	Physical position of start of archive file on media (`tar`(1) file) and file offset within the archive file in hexadecimal.
8	File system name.
9	Inode number and generation number. The generation number is an additional number used in addition to the inode number for uniqueness since inode numbers get re-used.
10	Length of file if file is written on only 1 volume. Length of section if file is written on multiple volumes.
11	Path and name of file relative to the file system's mount point.
12	Type of file, as follows: • `d` for directory. • `f` for regular file. • `l` for symbolic link. • `R` for removable media file. • `I` for segment index. • `S` for data segment.
13	Section of an overflowed file or segment. If the file is not overflowed or segmented, this number is 0.
14	Equipment Ordinal of the drive upon which the file was archived.

The archiver.cmd File Description

By default, the archiver runs whenever `ASMd` is started and a ASM or ASM-QFS file system is mounted. The default settings for the archiver are as follows:

- Archive all files to all available volumes.
- The archive age for all files is four minutes.
- The archive interval is 10 minutes.

It is likely that you will customize the actions of the archiver to meet the archiving requirements of your site. These actions are controlled by directives located in the archiver command file (`archiver.cmd`). The path name to this file is as follows:

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

If this file is not present, the archiver performs the default actions.

The archiver.cmd File

The `archiver.cmd` file consists of the following types of directives:

- General directives
- Archive set assignment directives
- Archive set directives
- VSN pool directives
- VSN association directives

The directives consist of lines of text read from the `archiver.cmd` file. Each directive line contains one or more fields separated by spaces or tabs. Any text that appears after the pound sign character (#) is treated as a comment and is not examined. Lines can be continued onto the next line by ending the line with a backslash (\).

Certain directives in the `archiver.cmd` file require you to specify a unit of time or a unit in bytes. To specify these units, use one of the following letters in Table 16 on page 88 as a suffix to the number that signifies the unit.

Table 16. The `archiver.cmd` File Directive Units

Time Suffix	Significance
`s`	Seconds.
`m`	Minutes. 60 seconds.
`h`	Hours. 3,600 seconds.
`d`	Days. 86,400 seconds.
`w`	Weeks. 604,800 seconds.
`y`	Years. 31,536,000 seconds.
`b`	Bytes.
`k`	Kilobytes. 2**10, or 1,024, bytes.
`M`	Megabytes. 2**20, or 1,048,576, bytes.
`G`	Gigabytes. 2**30, or 1,073,741,824, bytes.
`T`	Terabytes. 2**40, or 1,099,511,627,776, bytes.

An archiver.cmd File Example

Figure 6. shows a sample `archiver.cmd` file. The comments at the right indicate the various types of directives as listed in “The archiver.cmd File” on page 88.

The archiver checks the status of the `archiver.cmd` file once each minute. If the `archiver.cmd` file changes during archiver execution, the archiver stops scheduling archive copies and waits for copies in progress to be completed. It then reads the modified `archiver.cmd` file and restarts.

Note: If errors are found in the `archiver.cmd` file, the archiver logs the count of errors and displays the following message:

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

After displaying the previous message, the archiver waits to be restarted, which occurs when the `archiver.cmd` file changes or when the archiver receives an `arrun` or `arrestart` command from `samu`(1M).

Whenever you make changes to the `archiver.cmd` file, you should check for syntax errors using the `archiver`(1M) command. Specifying the `archiver`(1M) command as follows evaluates an `archiver.cmd` file against the current ASM or ASM-QFS system:

```
# archiver -lv
```

The preceding command lists all options and writes a listing of the `archiver.cmd` file, volumes, file system content, and errors to the standard output file (`stdout`). Errors prevent the archiver from running. The `archiver`(1M) command can be run on an archiver file-in-progress before moving the file to `/etc/opt/SUNWsamfs/archiver.cmd`. If you run the `archiver`(1M) command without an input file, archiver information is generated from `archiver.cmd`. If there is no `archiver.cmd` file, the system defaults are returned. For more information, see the `archiver`(1M) man page.

The following sections describe the directives in general. For more information on these directives, see the `archiver.cmd`(4) man page.

Figure 6. `archiver.cmd` **File Example**

```
interval = 30m                        # General directives
logfile = /var/opt/SUNWsamfs/archiver/archiver.log

fs = samfs1                           # Archive Set Assignments
no_archive tmp
work work
    1 1h
    2 3h
images images -minsize 100m
    1 1d
    2 1w
samfs1_all  .
    1 1h
    2 1h

fs = samfs2                           # Archive Set Assignments
no_archive tmp
system . -group sysadmin
    1 30m
    2 1h
samfs2_all  .
    1 10m
    2 2h
params                                # Archive Set Directives
allsets -drives 2
images.1 -join path -sort size
endparams
vsns                                  # VSN Associations
samfs1.1   mo      optic-2A
samfs1.2   lt      TAPE01
work.1     mo      optic-[3-9][A-Z]
work.2     lt      .*
images.1   lt      TAPE2[0-9]
images.2   lt      TAPE3[0-9]

samfs1_all.1       mo.*

samfs1_all.2       lt.*

samfs2.1   mo      optic-2A

samfs2.2   lt      TAPE01

system.1   mo      optic08a optic08b

system.2   lt      ^TAPE4[0-1]

samfs2_all.1       mo.*

samfs2_all.2       lt.*

endvsns
```

The archiver.cmd Directives

The following sections explain the `archiver.cmd` directives. They are as follows:

- "Global Archiving Directives" on page 91
- "Directives to Control Archiving for a Specific File System" on page 96
- "Archive Set Assignment Directive" on page 97
- "Archive Copy Directives" on page 103
- "Archive Set Parameters" on page 106
- "VSN Association Directives" on page 117
- "VSN Pools Directives" on page 119

Global Archiving Directives

General directives control the overall archiver operation. General directives in an `archiver.cmd` file can be identified either by the equal sign (=) in the second field or by the absence of additional fields. These directives allow you to optimize archiver operations for your site's configuration.

Global directives must be specified prior to any `fs=` directives in your `archiver.cmd` file. The `fs=` directives are those that pertain to specific file systems. The archiver issues a message if it detects a global directive after an `fs=` directive.

The interval Directive: Specifying an Archive Interval

The archiver executes periodically to examine the status of all mounted ASM and ASM-QFS file systems. The timing is controlled by the archive interval. The *archive interval* is the time between scan operations on each file system. To change the time, use the `interval` directive. This directive has the following format:

```
interval=time
```

The default interval is 10 minutes. If the archiver receives the `samu`(1M) utility's `:arrun` command, it begins scanning all file systems immediately.

If the `hwm_archive` mount option is set for the file system, the archive interval can be shortened automatically. This mount option specifies that the archiver commence its scan when the file system is filling up and the high water mark is crossed. The `high=`*percent* mount option sets the high water mark for the file system.

For more information on specifying the archive interval, see the `archiver.cmd`(4) man page. For more information on setting mount options, see the `mount_samfs`(1M) man page.

The bufsize Directive: Setting the Archiver Buffer Size

By default, a file being archived is copied to archive media using a memory buffer. You can use the `bufsize` directive to specify a nondefault buffer size and, optionally, to lock the buffer. These actions can improve performance, and you can experiment with different *buffer_size* values.

This directive has the following format:

```
bufsize=media buffer_size [ lock ]
```

where:

media	Specify the archive media type from the list on the `mcf`(4) man page.
buffer_size	Specify a number from 2 through 32. The default is 4. This value is multiplied by the *dev*`_blksize` value for the media type, and the resulting buffer size is used. The *dev*`_blksize` can be specified in the `defaults.conf` file. For more information on this file, see the `defaults.conf`(4) man page.
`lock`	The `lock` argument indicates whether or not the archiver should use locked buffers when making archive copies. If `lock` is specified, the archiver sets file locks on the archive buffer in memory for the duration of the `sam-arcopy`(1M) operation. This avoids the overhead of locking and unlocking the buffer for each I/O request and can result in a reduction in system CPU time. The `lock` argument should be specified only on large systems with large amounts of memory. Insufficient memory can cause an out-of-memory condition. The `lock` argument is effective only if direct I/O is enabled for the file being archived. By default, `lock` is not specified and the file system sets the locks on all direct I/O buffers, including those for archiving. For more information on enabling direct I/O, see the `setfa`(1) man page, the `sam_setfa`(3) library routine man page, or the `-O forcedirectio` option on the `mount_samfs`(1M) man page.

For example, this directive can be specified in the `archiver.cmd` file in a line like the following:

```
bufsize=od 7 lock
```

The `bufsize` and `lock` can also be set for an individual archive set. For more information on the preceding directives, see “Archive Set Parameters” on page 106.

The drives Directive: Controlling the Number of Drives Used for Archiving

By default, the archiver uses all of the drives in an automated library for archiving. To limit the number of drives in an automated library used by the archiver, use the drives directive.

This directive has the following format:

```
drives=auto_lib count
```

where:

auto_lib	The Family Set name of the automated library as defined in the `mcf` file.
count	The number of drives to be used for archiving activities.

The archmax Directive: Controlling the Size of Archive Files

The `archmax` directive specifies the maximum size of an archive file. User files are combined to form the archive file. No more user files are added to the archive file after the *target_size* is met. Large user files are written in a single archive file.

The maximum size of an archive file is media-dependent. By default, archive files written to optical disks are no larger than 5 megabytes. The default maximum archive file size for tapes is 512 megabytes.

To change the defaults, use the following directive:

```
archmax=media target_size
```

There are advantages and disadvantages to setting large or small sizes for archive files. For example, if you are archiving to tape and `archmax` is set to a large size, the tape drive stops and starts less often. However, when writing large archive files, there is the possibility that when an end-of-tape is reached prematurely, a large amount of tape can be wasted. As a rule, `archmax` should not be set to more than 5 percent of the media capacity. For example, the following `archmax` directive can be used for a 20 gigabyte tape:

```
archmax=sg 1G
```

The `archmax` directive can also be set for an individual archive set.

The ovflmin Directive: Controlling Volume Overflow

Volume overflow is the process of allowing archived files to span multiple volumes. For more information on volume overflow, see “Volume Overflow” on page 272.

Before using volume overflow, make sure that you understand the concept. Use volume overflow with caution only after thoroughly assessing the effect on your site. Disaster recovery and recycling are much more difficult with files that span volumes.

The archiver controls volume overflow through the `ovflmin` directive. The `ovflmin` directive specifies the minimum size file that is allowed to overflow a volume. By default, volume overflow is disabled.

This directive has the following format:

```
ovflmin = media minimum_file_size
```

where:

media	The media type. For a list of valid media types, see the `mcf`(4) man page.
minimum_file_size	Specify the minimum size of the file to overflow

For example, assume that many files exist with a length that is a significant fraction (say 25 percent) of an `mo` media cartridge. These files partially fill the volumes and leave unused space on each volume. To get better packing of the volumes, set `ovflmin` for `mo` media to a size slightly smaller than the size of the smallest file. The following directive sets it to 150 megabytes:

```
ovflmin=mo 150m
```

Note that enabling volume overflow in this example also causes two volumes to be loaded for archiving and staging the file.

The `ovflmin` directive can also be set for an individual archive set.

The wait Directive: Delaying Archiver Startup

The `wait` directive causes the archiver to wait for a start signal from `samu`(1M). When this signal is received, typical archiver operations are begun. By default, the archiver begins archiving when started by `ASMd`(1M). To delay archiving, use the `wait` directive. This directive has the following format:

```
wait
```

The `wait` directive can also be set for an individual file system.

The notify Directive: Renaming the Event Notification Script

The `notify` directive sets the name of the archiver's event notification script file to *filename*. This directive has the following format:

```
notify=filename
```

The default file name is `/opt/SUNWsamfs/sbin/archiver.sh`

This file is executed by the archiver to allow you to process various events in a site-specific manner. The script is called with a keyword for the first argument. The keywords are as follows: `emerg`, `alert`, `crit`, `err`, `warning`, `notice`, `info`, and `debug`.

Additional arguments are described in the default script.

The logfile Directive: Specifying An Archiver Log File

The archiver can produce a log file that contains information about each file that is archived, rearchived, or automatically unarchived. The log file is a continuous record of archival action. To specify a log file, use the `logfile` directive. This directive has the following format:

```
logfile=pathname
```

where:

pathname	Specify the absolute path name of the log file. By default, this file is not produced.

Example. Assume that you want to back up the archiver log file every day by copying the previous day's log file to an alternate location. This can be accomplished if you make sure that the copy is performed when the archiver log file is closed. In other words, you must not perform the copy operation while the archiver log file is open for a write operation. The steps you need to take are as follows:

1. Use the `mv`(1) command to move the archiver log file within UFS. This gives any `sam-arfind`(1M) or `sam-arcopy`(1M) operations time to finish writing to the archiver log file.
2. Use the `mv`(1) command to move the previous day's archiver log file to the ASM or ASM-QFS file system.

The `logfile` directive can also be set for an individual file system.

Directives to Control Archiving for a Specific File System

Directives specific to a particular file system can be included in the `archiver.cmd` file after the general directives. These directives specify actions to be taken only for individual file systems.

The fs Directive: Specifying the File System

By default, archiving controls apply to all file systems. However, you can confine some controls to an individual file system. To specify an individual file system, use the `fs` directive. This directive has the following format:

```
fs=fsname
```

where:

fsname Specify the file system name as defined in the `mcf` file.

The general directives and archive set association directives that occur after these directives apply only to the specified file system until another `fs=` directive is encountered. For instance, you can use this directive to specify a different log file for each file system.

Other File System Directives

Several directives can be specified both as global directives for all file systems and as directives that are specific to one file system. Their effects are the same regardless of where they are specified. These directives are as follows:

- The `interval` directive. For more information on this directive, see "The interval Directive: Specifying an Archive Interval" on page 91.
- The `logfile` directive. For more information on this directive, see "The logfile Directive: Specifying An Archiver Log File" on page 95.
- The `wait` directive. For more information on this directive, see "The wait Directive: Delaying Archiver Startup" on page 95.

Archive Set Assignment Directive

By default, files are archived as part of the archive set named for the file system. However, you can specify archive sets to include files that share similar characteristics. If a file does not match one of the specified archive sets, it is archived as part of the default archive set named for the file system.

The archive set membership directives assign files with similar characteristics to archive sets. The syntax of these directives is patterned after the `find`(1) command. Each archive set assignment directive has the following format:

```
archive_set_name path [search_criteria1 search_criteria2 … ] [file_attributes]
```

where:

archive_set_name	A site-defined name for the archive set. Must be the first field in the archive set assignment directive. An archive set name is usually indicative of the characteristics of the files belonging to the archive set. Archive set names are restricted to the letters in the alphabet, numbers, and the underscore character (_). No other special characters or spaces are allowed. The first character in the archive set name must be a letter. To prevent archiving for various files, specify `no_archive` as the *archive_set_name*.
path	A path relative to the mount point of the file system. This allows an archive set membership directive to apply to multiple ASM and ASM-QFS file systems. If the path is to include all of the files in a file system, use a period (.) for the path field. A leading slash (/) is not allowed in the path. Files in the directory specified by *path*, and its subdirectories, are considered for inclusion in this archive set.
search_criteria	Zero, one, or more *search_criteria* arguments can be specified. Search criteria can be specified to restrict the archive set according to file size, file ownership, and other factors. For information on possible *search_criteria* arguments, see the following sections.
file_attributes	Zero, one, or more *file_attributes* can be specified. These file attributes are set for files as the `sam-arfind` process scans a file system during archiving.

Example 1. The following are typical archive set membership directives:

```
hmk_files    net/home/hmk      -user hmk
datafiles    xray_group/data  -size 1M
system       .
```

Example 2. Archiving can be prevented by including files in an archive set named no_archive. The following lines prevent archiving of files in a tmp directory, at any level, and regardless of the directory in which the tmp directory resides within the file system:

```
fs = samfs1
no_archive tmp
no_archive . -name .*/tmp/
```

The following sections describe the *search_criteria* that can be specified.

File Size search_criteria: -minsize and -maxsize

The size of a file can be used to determine archive set membership using the -minsize *size* and -maxsize *size* characteristics. For *size*, specify an integer followed by one of the letters shown in Table 17..

Table 17. Size Suffixes

Letter	Meaning
b	Bytes
k	Kilobytes
M	Megabytes
G	Gigabytes
T	Terabytes

Example. The lines in this example specify that all files of at least 500 kilobytes, but less than 100 megabytes, belong to the archive set big_files. Files bigger than 100 megabytes belong to the archive set huge_files. The lines are as follows:

```
big_files  .  -minsize 500k  -maxsize 100M
huge_files .  -minsize 100M
```

Owner and Group search_criteria: -user and -group

The ownership and group affiliation can be used to determine archive set membership using the `-user` *name* and `-group` *name* characteristics. For example:

```
adm_set    .   -user sysadmin
mktng_set  .   -group marketing
```

All files belonging to user `sysadmin` belong to archive set `adm_set`, and all files with the group name of `marketing` are in the archive set `mktng_set`.

File Name search_criteria Using Pattern Matching: -name regex

The names of files that are to be included in an archive set can be specified by using regular expressions. The `-name` *regex* specification as a *search_criteria* specifies that any complete path matching the regular expression *regex* is a member of the archive set.

The *regex* argument follows the conventions as outlined in the `regexp`(5) man page. Note that regular expressions do not follow the same conventions as UNIX wildcards.

Internally, all files beneath the selected directory are listed (with their specified paths relative to the mount point of the file system) and passed along for pattern matching. This allows you to create patterns in the `-name` *regex* field to match both file names and path names.

Examples

1. The following directive restricts files in the archive set `images` to those files ending with `.gif`:

```
images  . -name \.gif$
```

2. The following directive selects files that start with the characters `GEO`:

```
satellite  . -name /GEO
```

3. You can use regular expressions with the `no_archive` archive set. The following specification prevents any file ending with `.o` from being archived:

```
no_archive  . -name \.o$
```

4. Assume that your `archiver.cmd` file contains the following lines:

```
# File selections.
fs = samfs1
     1 1s
     2 1s
no_archive share/marketing -name fred\.*
```

With this `archiver.cmd` file, the archiver does not archive `fred.*` in the user directories or subdirectories. Archiving occurs for files as follows:

- The following files are not archived:

```
/sam1/share/marketing/fred.anything
/sam1/share/marketing/first_user/fred.anything
/sam1/share/marketing/first_user/first_user_sub/
fred.anything
```

- The following files are archived:

```
/sam1/fred.anything
/sam1/share/fred.anything
/sam1/testdir/fred.anything
/sam1/testdir/share/fred.anything
/sam1/testdir/share/marketing/fred.anything
/sam1/testdir/share/marketing/second_user/fred.anything
```

5. Assume that your `archiver.cmd` file contains the following lines:

```
# File selections.
fs = samfs1
     1 1s
     2 1s
no_archive share/marketing -name ^share/marketing/[^/]*/
fred\.
```

This `archiver.cmd` file does not archive `fred.*` in the user home directories. This archives `fred.*` in the user subdirectories and in the directory `share/marketing`. In this case, the user home directories happen to be `first_user`. This example takes anything as a user’s home directory from `share/marketing/` until the next slash character (`/`). Archiving occurs for files as follows:

- The following files are not archived:

```
/sam1/share/marketing/first_user/fred.anything
```

- The following files are archived:

```
/sam1/share/fred.anything
/sam1/share/marketing/fred.anything
/sam1/share/marketing/first_user/first_user_sub/
fred.anything
/sam1/fred.anything
/sam1/testdir/fred.anything
/sam1/testdir/share/fred.anything
/sam1/testdir/share/marketing/fred.anything
/sam1/testdir/share/marketing/second_user/fred.anything
/sam1/testdir/share/marketing/second_user/sec_user_sub/
fred.any
```

Release and Stage file_attributes: -release and -stage

The release and stage attributes associated with files within an archive set can be set using the `-release` and `-stage` options, respectively. Both of these settings override stage or release attributes that might have been set previously by a user. If these attributes are set after the file is archived, the settings are acknowledged the next time the file is archived.

The `-release` option has the following format:

```
-release attributes
```

The *attributes* for the `-release` directive follow the same conventions as the `release`(1) command and are as shown in Table 18..

Table 18. The `-release` Option

attributes	**Meaning**
`a`	Release the file following the completion of the first archive copy.
`n`	Never release the file.
`p`	Partially release the file's disk space.

The `-stage` option has the following format:

```
-stage attributes
```

The *attributes* for the `-stage` directive follow the same conventions as the `stage`(1) command and are as shown in Table 19..

Table 19. The `-stage` Option

attributes	**Meaning**
`a`	Associative stage the file.
`n`	Never stage the file.

The following example shows how you can use file name specifications and file attributes in order to partially release Macintosh resource directories:

```
MACS  . -name .*/\.rscs/  -release p
```

Archive Set Membership Conflicts

Sometimes the choice of path and other file characteristics for inclusion of a file in an archive set results in ambiguous archive set membership. These situations are resolved in the following manner:

1. The membership definition occurring first in the archive set is chosen.
2. Membership definitions local to a file system are chosen before any globally defined definitions.
3. A membership definition that exactly duplicates a previous definition is noted as an error.

As a consequence of these rules, more restrictive membership definitions should be placed earlier in the directive file.

When controlling archiving for a specific file system (using the `fs=`*fsname* directive), directives are evaluated local to the file system level before being evaluated globally. Thus, files can be assigned to a local archive set (including the `no_archive` archive set) instead of being assigned to a global archive. This has implications when setting global archive set assignments such as `no_archive`.

For example, assume that the following resides in an `archiver.cmd` file:

```
no_archive   . -name .*\.o$
fs = samfs1
    allfiles  .
fs = samfs2
    allfiles  .
```

It appears that the administrator did not intend to archive any of the `.o` files across both file systems. However, because the local archive set assignment `allfiles` is evaluated before the global archive set assignment `no_archive`, the `.o` files in the `samfs1` and `samfs2` file systems are archived.

To ensure that no `.o` files are archived in both file systems, the following directives can be used:

```
fs = samfs1
    no_archive   . -name .*\.o$
    allfiles  .
fs = samfs2
    no_archive   . -name .*\.o$
    allfiles  .
```

Archive Copy Directives

If you do not specify archive copies, a single archive copy is made for files in the archive set. By default, this copy is made when the archive age of the file is four minutes. If you require more than one archive copy, all copies, including the first, must be specified using archive copy directives.

The archive copy directives begin with a digit. This digit (`1`, `2`, `3`, or `4`) is the copy number. The digit is followed by one or more arguments that specify archive characteristics for that copy.

The archive copy directives must appear immediately after the archive set assignment directive to which they pertain. Each archive copy directive has the following format:

```
copy_number [ -release | -norelease ] [archive_age]
[unarchive_age]
```

The following sections describe the archive copy directive arguments.

Releasing Disk Space After Archiving: -release

You can specify that the disk space for files be automatically released after an archive copy is made by using the `-release` directive after the copy number. This option has the following format:

```
-release
```

In this example, files with the group `images` are archived when their archive age reaches 10 minutes. After archive copy 1 is made, the cache disk space is released.

```
ex_set . -group images
    1 -release 10m
```

Delaying Disk Space Release: -norelease

You might not want to release disk space until multiple archive copies are completed. The `-norelease` option prevents the automatic release of disk cache until all copies marked with `-norelease` are made. This option has the following format:

```
-norelease
```

The following example specifies an archive set named `vault_tapes`. Two copies are created, but the disk cache associated with this archive set is not released until both copies are made. This scenario can be used at a site that requires online access to files before creating offsite vault tapes.

```
vault_tapes
    1 -norelease 10m
    2 -norelease 30d
```

Note that the `-norelease` specification on a single copy has no effect on automatic releasing because the file cannot be released until it has at least one archive copy. Also, the `-norelease` and `-release` specifications are mutually exclusive.

Setting the Archive Age

You can set the archive age for files by specifying the archive age as the next field on the directive. The archive age can be specified with the suffix characters shown in Table 20..

Table 20. Age Suffixes

Time Suffix	Meaning
s	Seconds
m	Minutes
h	Hours
d	Days
w	Weeks
y	Years

In the following example, the files in directory `data` are archived when their archive age reaches one hour:

```
ex_set data
    1 1h
```

Unarchiving Automatically

If you specify more than one archive copy of a file, it is possible to unarchive all but one of the copies automatically. This might occur when the files are archived to various media using various archive ages.

The following example specifies the unarchive age:

```
ex_set home/users
    1 6m 10w
    2 10w
    3 10w
```

The first copy of the files in the path `home/users` is archived six minutes after modification. When the files are 10 weeks old, second and third archive copies are made. The first copy is then unarchived.

For more ways to control unarchiving, see “Controlling Unarchiving” on page 111.

Specifying More Than One Copy for Metadata

If more than one copy of metadata is required, copy definitions can be placed in the directive file immediately after an `fs=` directive. For example:

```
fs = samfs7
    1 4h
    2 12h
```

In this example, copy 1 of the metadata for the `samfs7` file system is made after four hours and a second copy is made after 12 hours.

File system metadata includes changes to path names in the file system. For this reason, if you have frequent changes to directories, new archive copies are created. This results in frequent loads of the volumes specified for metadata.

Archive Set Parameters

The archive set parameters section of the `archiver.cmd` file begins with the `params` directive and ends with the `endparams` directive. The format for directives for an archive set is as follows:

```
params
archive_set_name.copy_number [ -param1 -param2 …]
.
.
.
endparams
```

The pseudo archive set `allsets` provides a way to set default archive set directives for all archive sets. All `allsets` directives must occur before those for actual archive set copies. Parameters set for individual archive set copies override parameters set by `allsets` directives. For more information on the `allsets` archive set, see the `archiver.cmd`(4) man page.

All archive set processing parameters are described in this section, with the exception of the `-disk_archive` parameter. For information on the `-disk_archive` parameter, see “Disk Archiving” on page 120.

Setting the Archiver Buffer Size

By default, a file being archived is stored in memory in a buffer prior to writing the file to archive media. You can use the `-bufsize` parameter to specify a nondefault buffer size. These actions can improve performance, and you can experiment with various *buffer_size* values.

This parameter has the following format:

```
-bufsize=buffer_size
```

For *buffer_size*, specify a number from 2 through 32. The default is 4. This value is multiplied by the *dev*_`blksize` value for the media type, and the resulting buffer size is used. The *dev*_`blksize` can be specified in the `defaults.conf` file. For more information on this file, see the `defaults.conf`(4) man page.

For example, this parameter can be specified in the `archiver.cmd` file in a line such as the following:

```
myset.1 -bufsize=6
```

The equivalent of this directive can also be specified on a global basis by specifying the `bufsize=`*media buffer_size* directive. For more information on this topic, see “The bufsize Directive: Setting the Archiver Buffer Size” on page 92.

Specifying Archive Buffer Locks

By default, a file being archived is stored in memory in a buffer prior to writing the file to archive media. If direct I/O is enabled, you can use the `-lock` parameter to lock this buffer. This action can improve performance, and you can experiment with this parameter.

This parameter has the following format:

```
-lock
```

The `-lock` parameter indicates whether or not the archiver should use locked buffers when making archive copies. If `-lock` is specified, the archiver sets file locks on the archive buffer in memory for the duration of the `sam-arcopy`(1M) operation. This avoids paging the buffer, and it can improve performance.

The `-lock` parameter should be specified only on large systems with large amounts of memory. Insufficient memory can cause an out-of-memory condition.

The `-lock` parameter is effective only if direct I/O is enabled for the file being archived. By default, `-lock` is not specified and the file system sets the locks on all direct I/O buffers, including those for archiving. For more information on enabling direct I/O, see the `setfa`(1) man page, the `sam_setfa`(3) library routine man page, or the `-O forcedirectio` option on the `mount_samfs`(1M) man page.

For example, this parameter can be specified in the `archiver.cmd` file in a line such as the following:

```
yourset.3 -lock
```

You can also specify the equivalent of this parameter on a global basis by specifying the `lock` argument to the `bufsize=`*media buffer_size* [`lock`] directive. For more information on this topic, see “The bufsize Directive: Setting the Archiver Buffer Size” on page 92.

Assigning Multiple Drives to an Archive Set

By default, the archiver usually uses only one media drive to archive files in an archive set. When an archive set has a many files or large files, it can be advantageous to use more than one drive. You can specify this by using the `-drives` parameter. This parameter has the following format:

```
-drives number
```

For *number*, specify the number of drives to be used for archiving the specified archive set.

For example:

```
huge_files.2 -drives 2
```

When the total size of the files in archive set `huge_files.2` is equal to or greater than two times `drivemin` for the media, two drives are used to archive the files.

Splitting Archive Requests Among Drives

You can use the `-drivemin` *min_size* parameter in conjunction with the `-drives` directive to set the minimum size for splitting archive requests among drives. For example, you can use the `-drivemin` parameter if you want to divide an archive request among drives, but you want to avoid tying up all the drives with small archive requests. This might apply to operations that use very large files.

The default for the `-drivemin` parameter is the setting for the `-archmax` parameter. The `-archmax` parameter's default setting is the *target_size* for the specific volume being used.

The `-drivemin` directive has the following format:

```
-drivemin min_size
```

The `-drivemin` parameter sets the minimum size of the multiple drives for the archive set to *min_size*. When you use the `-drives` parameter, multiple drives are used only if data that is more than the *min_size* is to be archived at once. The number of drives to be used in parallel is the lesser of *arch_req_total_size*/*min_size* and the number of drives specified by the `-drives` parameter.

An archive request is evaluated against both the `-drives` and `-drivemin` parameters, as follows:

- If an archive request is less than *min_size*, only one drive is used to write an archive request.
- If an archive request is larger than *min_size*, the archive request is evaluated against *min_size* and the appropriate number of drives is scheduled up to the full number of drives specified.
- If *min_size* is zero, an attempt is made to split among the full number of drives specified.

For example, assume that you are splitting an archive set named big_files over five drives. Depending on its size, this archive set could be split as shown in Table 21..

Table 21. Archive Set Example Split

Archive Set Size	Number of Drives
< 20 gigabytes	1
≥ 20 gigabytes to < 30 gigabytes	2
≥ 30 gigabytes to < 40 gigabytes	3
≥ 40 gigabytes to < 50 gigabytes	4
≥ 50 gigabytes	5

The following line would be used in the /etc/opt/SUNWsamfs/archiver.cmd file for this example:

```
params
bigfiles.1 -drives 5 -drivemin 10G
endparams
```

Specifying Recycling

The recycling process allows you to reclaim space on archive volumes that is taken up by expired archive images. By default, no recycling occurs, but if you want to recycle, you can specify recycling directives in the archiver.cmd file to control this process.

For more information on the recycling directives supported in the archiver.cmd file, see “Recycling” on page 177.

Associative Archiving

Associative archiving is used if you specify the -join path parameter. Associative archiving is useful if you want an entire directory to be archived to one volume and you know that the archive file can physically reside on only one volume. Otherwise, if you want to keep directories together, use the -sort path parameter to keep the files contiguous.

When an archive file is written to a volume, files are written to an archive file to efficiently pack the volume with user files. Subsequently, when accessing files from the same directory, you can experience delays as the stage process repositions through a volume to read the next file. To alleviate delays, you can archive files from the same directory paths contiguously within an archive file. The process of associative archiving overrides the space efficiency algorithm to archive files from the same directory together. The -join path parameter allows these files to be archived contiguously within an archive set copy.

Associative archiving is useful when the file content does not change but you want to access the group of files together at the same time all the time. For example, you might use associative archiving at a hospital for accessing medical images. Images associated with the same patient can be kept in a directory and the doctor might want to access those images together at one time. These static images can be more efficiently accessed if you archive them contiguously based on their directory location. For example:

```
patient_images.1 -join path
```

Note: The `-join path` parameter writes data files from the same directory to the same archive file. If there are many directories with a few small files, the archiver creates many small archive files. These small, discrete archive files slow the write performance of the system because the data files are relatively small compared to the `tar`(1) header for each archive file. This can impair performance when writing to high-speed tape drives.

Also, because the `–join path` parameter specifies that all the files from the same directory be archived on a single volume, it is possible that a group of files might not fit on any available volume. In this case, the files are not archived until more volumes are assigned to the archive set. It is also possible that the group of files to be archived is so large that it can never fit on a single volume. In such a case, the files are never archived.

For most applications, using the `-sort path` parameter is more efficient than using `-join path` if the more restrictive operation of `-join path` is not a requirement.

It is also possible to sort files within an archive set copy by age, size, or path. The `age` and `size` arguments are mutually exclusive. To sort an archive set, use the `-sort` parameter with the argument `age` or `size`, as follows:

```
cardiac.1 –sort path
cardiac.2 -sort age
catscans.3 -sort size
```

The first line forces the archiver to sort an archive request by path name. The second example line forces the archiver to sort an archive set copy called `cardiac.2` by the age of the file, youngest to oldest. The third line forces the archive set copy called `catscans` to be sorted by the size of the file, largest to smallest.

Controlling Unarchiving

Unarchiving is the process by which archive entries for files or directories are deleted. By default, files are never unarchived. Files are unarchived based on the time since last access. All frequently accessed data can be stored on a fast media, such as disk, and all older, infrequently accessed data can be stored on tape.

Example 1

Assume that your ASM `archiver.cmd` file contains the following lines:

```
arset1 dir1
   1    10m    60d
   2    10m
   3    10m
vsns
arset1.1    mo       OPT00[0-9]
arset1.2    lt       DLTA0[0-9]
arset1.3    lt       DLTB0[0-9]
```

If a file controlled by the preceding `archiver.cmd` file is accessed frequently, it remains on disk all the time, even if it is older than 60 days. The copy 1 information is removed only if the file is not accessed for 60 days.

If the copy 1 information is removed (because the file was not accessed for 60 days) and someone stages the file from copy 2, it is read from tape. After the file is back online, the archiver makes a new copy 1 on disk and the 60-day access cycle starts all over again. The ASM archiver regenerates a new copy 1 if the file is accessed again.

Example 2

Assume that a patient is in the hospital for four weeks. During this time, all of this patient's files are on fast media (copy 1=`mo`). After four weeks, the patient is released from the hospital. If no data has been accessed for this patient for up to 60 days after the patient is released, the copy 1 entry in the inode is unarchived, and only copy 2 and copy 3 entries are available. The volume can now be recycled in order to make room for more current patients without having to increase the disk library. If the patient comes back to the hospital after six months for a checkup, the first access of the data is from tape (copy 2). Now the archiver automatically creates a new copy 1 on disk to ensure that the data is back on the fast media during the checkup, which could take several days or weeks.

Controlling How Archive Files are Written

By default, the archiver writes a tape mark, an EOF label, and two more tape marks between archive files. When the next archive file is started, the driver backs up to the position after the first tape mark, causing a loss of performance. The `-tapenonstop` parameter directs the archiver to write only the initial tape mark. In addition, if the `-tapenonstop` parameter is specified, the archiver enters the archive information at the end of the copy operation.

For more information on the `-tapenonstop` parameter, see the `archiver.cmd`(4) man page.

Reserving Volumes

By default, the archiver writes archive set copies to any volume specified by a regular expression as described in the volume associations section of the `archiver.cmd` file. However, it is sometimes desirable for archive set volumes to contain files from only one archive set. The process of reserving volumes can be used to satisfy this data storage requirement.

Note: The `-reserve` parameter reserves a volume for exclusive use by one archive set. A site that uses reserved volumes is likely to incur more cartridge loads and unloads.

The `-reserve` parameter reserves volumes for an archive set. When the `-reserve` parameter is set and a volume has been assigned to an archive set copy, the volume identifier is not assigned to any other archive set copy, even if a regular expression matches it.

As volumes are selected for use by an archive set, a reserved name is assigned to the volume. The reserved name is a unique identifier that ties the archive set to the volume.

The format for the `-reserve` parameter is as follows:

```
-reserve keyword
```

The *keyword* specified depends on the form you are using. The possible forms are archive set form, owner form, and file system form, as follows:

- Archive set form. This form uses the `set` *keyword*, as follows: `-reserve set`
- Owner form. This form uses one of the following *keywords*: `dir`, `user`, or `group`. The formats for these are as follows:

```
-reserve dir
-reserve user
-reserve group
```

  The preceding three owner forms are mutually exclusive. That is, only one of the three owner forms can be used on an archive set and copy.

- File system form. This form uses the `fs` *keyword*, as follows: `-reserve fs`

In the `archiver.cmd` file, you can specify a `-reserve` parameter for one, two, or all three possible forms. The three forms can be combined and used together in an archive set parameter definition.

For example, in the following `archiver.cmd` file fragment, the line that begins with `arset.1` creates a reserved name based upon an archive set, a group, and the file system:

```
params
arset.1 -reserve set -reserve group -reserve fs
endparams
```

The information regarding reserved volumes is stored in the library catalog. The lines within the library catalog contain the media type, the VSN, the reserve information, and the reservation date and time. The reserve information includes the archive set component, path name component, and file system component, separated by slashes (`//`).

These slashes are *not* indicative of a path name; they are merely separators for displaying the three components of a reserved name. As the following fragment from an example library catalog shows, the lines describing reserved volumes begin with `#R` characters:

```
     6  00071  00071 lt     0xe8fe   12 9971464 1352412
0x6a000000 131072 0x
#     -il-o-b-----  05/24/00 13:50:02  12/31/69 18:00:00  07/
13/01 14:03:00
#R lt 00071 arset0.3// 2001/03/19 18:27:31
    10 ST0001 NO_BAR_CODE lt     0x2741    9 9968052 8537448
0x68000000 1310
#     -il-o-------  05/07/00 15:30:29  12/31/69 18:00:00  04/
13/01 13:46:54
#R lt ST0001 hgm1.1// 2001/03/20 17:53:06
    16 SLOT22 NO_BAR_CODE lt     0x76ba    6 9972252 9972252
0x68000000 1310
#     -il-o-------  06/06/00 16:03:05  12/31/69 18:00:00  07/
12/01 11:02:05
#R lt SLOT22 arset0.2// 2001/03/02 12:11:25
```

Note that some lines in the preceding code fragment have been truncated to fit on the page.

One or more of the reserve information fields can be empty, depending on the options defined in the `archiver.cmd` file. The date and time indicate when the reservation was made. A reservation line is appended to the file for each volume that is reserved to an archive set during archiving.

You can display the reserve information by using the `samu`(1M) utility's `v` display or by using the `archiver`(1M) or `dump_cat`(1M) commands in one of the following formats:

```
archiver -lv
dump_cat -V catalog_name
```

The following formats illustrate each form showing the parameter, keywords, and examples of reserved names assigned to volumes.

- Archive set form. The `set` keyword activates the archive set component in the reserved name as shown in Table 22..

Table 22. Archive Set Form Examples

Directive and Keyword	Reserved Name Examples
`-reserve set`	`users.1//`
	`Data.1//`

For example, in the following `archiver.cmd` file fragment, the line that begins with the `allsets` archive set name sets reserve by archive set for all archive sets:

```
params
allsets -reserve set
endparams
```

- Owner form. The `dir`, `user`, and `group` keywords activate the owner component in the reserved name. The `dir`, `user`, and `group` keywords are mutually exclusive. The `dir` keyword uses the directory path component immediately following the path specification of the archive set definition. The `user` and `group` keywords are self-explanatory. Examples are shown in Table 23..

Table 23. Owner Set Form Examples

Directive and Keyword	Reserved Name Examples
`-reserve dir`	`proj.1/p105/`
	`proj.1/p104/`
`-reserve user`	`users.1/user5/`

Table 23. Owner Set Form Examples

Directive and Keyword	Reserved Name Examples
	`users.1/user4/`
`-reserve group`	`data.1/engineering/`

Note: The `-reserve` parameter is intended to reserve a volume for exclusive use by one archive set. Many directories with a few small files cause many small archive files to be written to each reserved volume. These small discrete archive files slow the performance of the system because data files are relatively small compared to the `tar`(1) header for each archive file.

- File system form. The `fs` keyword activates the file system component in the Reserved Name. Examples are shown in Table 24..

Table 24. File System Form Examples

Directive and Keyword	Reserved Name Examples
`-reserve fs`	`proj.1/p103/samfs1`
	`proj.1/p104/samfs1`

A complete archive example using reserved volumes is presented at the end of this chapter as “Example 4” on page 136.

The archiver records volume reservations in the library catalog files. A volume is automatically unreserved when it is relabeled because the archive data has been effectively erased.

You can also use the `reserve`(1M) and `unreserve`(1M) commands to reserve and unreserve volumes. For more information on these commands, see the `reserve`(1M) and `unreserve`(1M) man pages.

Setting Archive Priorities

The ASM and ASM-QFS file systems offer a configurable priority system for archiving files. Each file is assigned a priority computed from properties of the file and priority multipliers that can be set for each archive set in the `archiver.cmd` file. Properties include online/offline, age, number of copies made, and size.

By default, the files in an archive request are not sorted and all property multipliers are zero. This results in files being archived in first found, first archived order. For more information on priorities, see the `archiver`(1M) and `archiver.cmd`(4) man pages.

You can control the order in which files are archived by setting priorities and sort methods. The following are examples of priorities that you can set:

- Select the `priority` sort method to archive files within an archive request in priority order.
- Change the `archive_loaded` priority to reduce media loads.
- Change the `offline` priority to cause online files to be archived before offline files.
- Change the `copy#` priorities to make archive copies in copy order.

Table 25. lists the archive priorities.

Table 25. Archive Priorities

Archive Priority	Definition
`-priority age`	Archive age property multiplier
`-priority archive_immediate`	Archive immediate property multiplier
`-priority archive_overflow`	Multiple archive volumes property multiplier
`-priority archive_loaded`	Archive volume loaded property multiplier
`-priority copy1`	Copy 1 property multiplier
`-priority copy2`	Copy 2 property multiplier
`-priority copy3`	Copy 3 property multiplier
`-priority copy4`	Copy 4 property multiplier
`-priority copies`	Copies made property multiplier
`-priority offline`	File offline property multiplier
`-priority queuewait`	Queue wait property multiplier
`-priority rearchive`	Rearchive property multiplier
`-priority reqrelease`	Reqrelease property multiplier
`-priority size`	File size property multiplier
`-priority stage_loaded`	Stage volume loaded property multiplier
`-priority stage_overflow`	Multiple stage volumes property multiplier

VSN Association Directives

The VSN associations section of the `archiver.cmd` file assigns volumes to archive sets. This section starts with a `vsns` directive and ends with an `endvsns` directive.

Collections of volumes are assigned to archive sets by directives of the following form:

```
archive_set_name.copy_num media_type vsn_expr ... [ -pool
vsn_pool_name ... ]
```

where:

archive_set_name	A site-defined name for the archive set. Must be the first field in the archive set assignment directive. An archive set name is usually indicative of the characteristics of the files belonging to the archive set. Archive set names are restricted to the letters in the alphabet, numbers, and the underscore character (_). No other special characters or spaces are allowed. The first character in the archive set name must be a letter.
copy_num	A digit that is followed by one or more arguments that specify archive characteristics for that copy. Archive copy directives begin with a digit. This digit (`1`, `2`, `3`, or `4`) is the copy number.
media_type	The media type. For a list of valid media types, see the `mcf`(4) man page.
vsn_expr	A regular expression. See the `regexp`(5) man page.
`-pool` *vsn_pool_name*	A named collection of VSNs.

An association requires at least three fields: the *archive_set_name* and *copy_number*, the *media_type*, and at least one volume. The *archive_set_name* and *copy_number* are connected by a period (.).

The following examples specify the same VSNs in different ways.

Example 1. The following example shows two lines of VSN specifications:

```
vsns
set.1  lt  VSN001 VSN002 VSN003 VSN004 VSN005
set.1  lt  VSN006 VSN007 VSN008 VSN009 VSN010
endvsns
```

Example 2. The following example shows a VSN specification that uses a backslash character (\) to continue a line onto a subsequent line:

```
vsns
set.1  lt  VSN001 VSN002 VSN003 VSN004 VSN005 \
 VSN006 VSN007 VSN008 VSN009 VSN010
endvsns
```

Example 3. The following example specifies VSNs using a regular expression in a shorthand notation:

```
vsns
set.1 lt VSN0[1-9] VSN10
endvsns
```

The volumes are noted by one or more *vsn_expression* keywords, which are regular expressions as described in the `regexp`(5) man page. Note that these regular expressions do not follow the same conventions as wildcards. In addition to a regular expression, you can also specify VSN pools from which volumes are to be selected. Pools are expressed with the `-pool` *vsn_pool_name* directive with a VSN association.

When volumes are needed by the archiver for the archive set, each volume of the selected media type in all automated libraries and manually mounted drives is examined to determine if it would satisfy any VSN expression. The first volume that fits an expression that contains enough space for the archive copy operation is selected. For example:

- The following directive specifies that files belonging to archive set `ex_set` for copy 1 be copied to media type `mo` using any of the twenty volumes with the name `optic20` through `optic39`:

```
ex_set.1 mo optic[2-3][0-9]
```

- The following directive copies files belonging to archive set `ex_set` for copy 2 to media type lt with any volume beginning with `TAPE`:

```
ex_set.2  lt  ^TAPE
```

If your ASM or ASM-QFS environment is configured to recycle by archive set, do not assign a VSN to more than one archive set.

Note: Make sure you assign volumes to the archive set for the metadata when setting up the `archiver.cmd` file. Each file system has an archive set with the same name as the file system. For more information on preserving metadata, see the `samfsdump`(1M) man page or see the *ASM, ASM-QFS, and ASM/QFS-Standalone Disaster Recovery Guide*.

VSN Pools Directives

The VSN pools section of the `archiver.cmd` file starts with a `vsnpools` directive and ends with either an `endvsnpools` directive or with the end of the `archiver.cmd` file. This section names a collection of volumes.

A *VSN pool* is a named collection of volumes. VSN pools are useful for defining volumes that can be available to an archive set. As such, VSN pools provide a useful buffer for assigning volumes and reserving volumes to archive sets.

You can use VSN pools to define separate groups of volumes for use by departments within an organization, users within a group, data types, and other convenient groupings. The pool is assigned a name, media type, and a set of volumes. A *scratch pool* is a set of volumes used when specific volumes in a VSN association are exhausted or when another VSN pool is exhausted. For more information on VSN associations, see “VSN Association Directives” on page 117.

If a volume is reserved, it is no longer available to the pool in which it originated. Therefore, the number of volumes within a named pool changes as volumes are used. You can view the VSN pools by entering the `archiver`(1M) command in the following format:

```
# archiver -lv | more
```

A VSN pool definition requires at least three fields separated by white space: the pool name, the media type, and at least one VSN. The syntax is as follows:

```
vsn_pool_name media_type vsn_expression
```

where:

vsn_pool_name	Specifies the VSN pool
media_type	The 2-character media type. For a list of valid media types, see the `mcf`(4) man page.
vsn_expression	Regular expression; there can be one or more *vsn_expression* argumemts. See the re`gcmp`(3G) man page.

The following example uses four VSN pools: `users_pool`, `data_pool`, `proj_pool`, and `scratch_pool`. If one of the three specific pools is out of volumes, the scratch pool VSNs are selected. The example is as follows:

```
vsnpools
users_pool   mo ^MO[0-9][0-9]
data_pool    mo ^DA.*
scratch_pool mo ^SC[5-9][0-9]
proj_pool    mo ^PR.*
endvsnpools
vsns
users.1     mo    -pool users_pool   -pool scratch_pool
data.1      mo    -pool data_pool    -pool scratch_pool
proj.1      mo    -pool proj_pool    -pool scratch_pool
endvsns
```

■ Disk Archiving

Archiving is the process of copying a file from online disk to archive media. Often, archive copies are written to volumes on magneto optical or tape cartridges in an automated library, but with disk archiving, online disk in a file system is used as archive media.

Disk archiving can be implemented so that the files are archived from one ASM or ASM-QFS file system to another file system on the same host computer system. Disk archiving can also be implemented so the source files are archived to another file system on a different Solaris system. When disk archiving is implemented using two host systems, the systems involved act as a client and a server. The *client system* is the system that hosts the source files. The *server system* is the destination system that hosts the archive copies.

The file system to which the archive files are written can be any UNIX file system. It does not have to be a ASM or ASM-QFS file system. If disk archive copies are written to a different server system, the server system must have at least one ASM or ASM-QFS file system installed upon it.

The archiver treats files archived to disk volumes the same as it treats files archived to volumes in a library. You can still make one, two, three, or four archive copies. If you are making multiple archive copies, one of the archive copies could be written to disk volumes while the others are written to removable media volumes. In addition, if you typically archive to disk volumes in a ASM or ASM-QFS file system, the archive file copies are themselves archived according to the `archiver.cmd` file rules in that file system.

The following list summarizes some of the similarities and differences between archiving to online disk and archiving to removable media:

- Unlike archive copies written to a magneto optical disk or to a tape, archive copies written to disk are not recorded in a catalog. In addition, archive files in disk volumes do not appear in the historian.

- If you are archiving to removable media volumes, you can begin archiving after the file system is mounted without changing any of the default values in the `archiver.cmd` file. If you are archiving to disk volumes, however, the `archiver.cmd` file must be edited prior to mounting the file system in order to define disk archive sets.
- Disk archiving does not rely on entries in the `mcf`(4) file. You need to specify the `-disk_archive` parameter in the `archiver.cmd` file, and you need to define disk volumes in `/etc/opt/SUNWsamfs/diskvols.conf`. This is an additional configuration file, and it is not needed if you are archiving to removable media volumes only.

  A `diskvols.conf` file must be created on the system upon which the source files reside. Depending on where the archive copies are written, this file also contains the following information:

  - If the archive copies are written to a file system on that same host system, the `diskvols.conf` file defines the VSNs and the paths to each VSN.
  - If the archive copies are written to a different Solaris system, the `diskvols.conf` file contains the host name of that server system. In this case, there must also be a `diskvols.conf` file on the server system that defines clients that are given permission to write to that system.

The following sections describe the files you need to configure to enable disk archiving.

Defining Disk VSNs

Disk volumes are defined in the `/etc/opt/SUNWsamfs/diskvols.conf` file. This is the file in which you define the disk volumes to be used for receiving archive copies. The following sections describe the `diskvols.conf` file and present guidelines on configuring disk archiving.

The diskvols.conf File

A `diskvols.conf` file must exist on the Solaris system that contains source files to be archived. If the source files are written to archive copies on another Solaris system, a client/server relationship exists, and there must be an additional `diskvols.conf` file on that server system.

Comments are permitted in the `diskvols.conf` file. A pound character (#) indicates a comment, and all text to the right of the # is ignored.

Continuation lines are permitted in the `diskvols.conf` file. To continue a line, put an apostrophe character (‘) at the end.

Each line in the `diskvols.conf` file is divided into the following fields:

- The `VSN Name` field contains a unique name for the disk archive volume. This name can be up to 31 characters long.
- The `Host Name` is an optional field. If you are archiving files from a client system to a server system, you must specify the host name of the server system. The `Host Name` field must be the name of the server host system upon which the file system to which the archive copies are written is mounted. The Solaris system to which the archive copies are written must have a ASM or ASM-QFS-QFS file system installed on it. If the `Host Name` field is specified, it must be followed by a colon character (:). If information is provided in the `Host Name` field, you must also create a `diskvols.conf` file on that server system.
- The `Path` field specifies the location (directory) to which archive copies are written. The specified directory structure must be in place before archiving can begin, and the file system in which this directory resides must be mounted. If the file system becomes unmounted, no archiving occurs. The `Path` should be specified relative to the mount point. For example, if archive copies are to be written to `vsns` directory in the `archivefs1` file system, you would specify `/archivefs1/vsns` in the `Path` field.

  StorageTek recommends that the destination directory be created with write permission granted only to `root`.

Example. The following `diskvols.conf` file resides on client system `pluto`:

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on pluto
# VSN Name     [Host Name:]Path
#
disk01                    /sam_arch1
disk02                    /sam_arch2/proj_1
disk03                mars:/sam_arch3/proj_3
```

In the preceding `diskvols.conf` file, VNS identified as `disk01` and `disk02` are written to the host system upon which the original source files reside. VSN `disk03` is written to a VSN on server system `mars`.

The following `diskvols.conf` file resides on server system `mars`:

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on mars
#
clients
pluto
endclients
```

The following is a fragment of the `archiver.cmd` file on `pluto`:

```
params
arset1.2 -disk_archive disk01
arset2.2 -disk_archive disk02
arset3.2 -disk_archive disk03
endparams
```

Configuration Guidelines

While there are no restrictions on where disk archive volumes can reside, it is recommended that the disk volumes reside on a disk other than the one upon which the original files reside. Preferably, the archive copies from a client system would be written to disk volumes on a server system. It is recommended that you make more than one archive copy and write to more than one type of archive media. For example, copy 1 could be archived to disk volumes, copy 2 to tape, and copy 3 to magneto-optical disk.

If you are archiving files to a file system on a server system, the archive files themselves can be archived to removable media cartridges in a library attached to the destination server.

Defining Disk Archive Sets

The `archiver.cmd` file on the system that hosts the source files must be edited to include information on disk archive sets.

The `-disk_archive` parameter in the `archiver.cmd` file defines a disk archive set. The archiver uses this parameter to maintain the file system hierarchy of the data as it is written to the archive disk's mount point. Like all archive set processing parameters, it must be specified between the `params` and `endparams` directives.

The format of this directive is as follows:

```
params
archive_set.copy_number -disk_archive VSN_Name
endparams
```

For *VSN_Name*, specify a VSN that is defined in the `diskvols.conf` file.

Note: When archiving to online disk, only a subset of the `archiver.cmd` directives are recognized for the disk archive set. These directives, which are used to define the archive set and to configure recycling, are as follows:

```
-disk_archive
-recycle_hwm
-recycle_ignore
```

-recycle_mailaddr
-recycle_mingain

For more information on the recycling directives, see the archiver.cmd(4) man page.

Example 1

In this example, file /sam1/testdir0/filea is in the archive set for arset0.1, and the archiver copies the content of /sam1/testdir0/filea to the destination path named /sam_arch1.

The diskvols.conf file for this is as follows:

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf
#
# VSN Name   [Host Name:]Path
#
disk01                  /sam_arch1
disk02                  /sam_arch12/proj_1
```

The archiver.cmd file contains the following lines that pertain to disk archiving:

```
.
.
.
params
arset0.1 -disk_archive disk01
endparams
.
.
.
```

The following shows output from the sls(1) command for file filea, which was archived to disk. In the example output, dk is the media type for disk archive media, and disk02 is the VSN:

```
# sls -D filea
mode: -rw-r-----  links:  1  owner: root      group: other
length:  5766596  inode:     988
archdone;
copy 1: ---- Jan 11 08:34     0.1    dk disk02
access:   Jan 11 08:26   modification:  Jan 11 08:26
changed:  Jan 11 08:26   attributes:    Jan 11 08:26
creation: Jan 11 08:26   residence:     Jan 11 08:26
```

Example 2

In this example, file /sam2/my_proj/fileb is on client host snickers in archive set arset0.1, and the archiver copies the content of this file to the destination path /sam_arch1 on server host mars.

The diskvols.conf file on snickers is as follows:

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on snickers
#
# VSN Name   [Host Name:]Path
#
disk01        mars:/sam_arch1
```

The diskvols.conf file on mars is as follows:

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on mars
#
clients
snickers
endclients
```

The directives in the archiver.cmd file that relate to this example are as follows:

```
.
.
.
params
arset0.1 -disk_archive disk01
endparams
.
.
.
```

Archiver Examples

All archiver examples in this section assume the following directory structure:

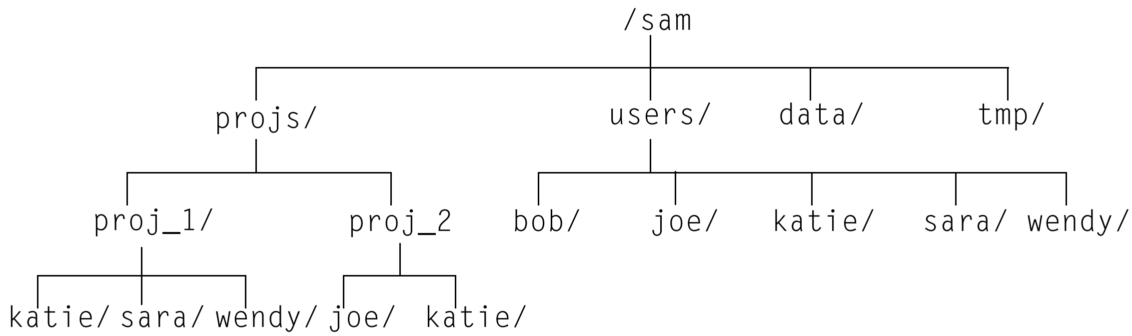

Figure 7. Archiver Example Directory Structure

Example 1

This example illustrates the action of the archiver when no `archiver.cmd` file is used. In this example, an ASM environment includes one file system, an optical automated library with two drives, and six cartridges. The example shows the output produced by the following command:

```
# archiver -lv
```

The following output shows that the default media selected by the archiver is type `mo`. Only the `mo` media are available:

```
Notify file: /opt/SUNWsamfs/sbin/archiver.sh

Archive media:
media:lt archmax: 512.0M Volume overflow not selected
media:mo archmax:   4.8M Volume overflow not selected
```

The following output indicates that the archiver uses two drives. The 12 volumes, storage capacity, and available space are listed.

```
Archive libraries:
Device:hp30 drives_available:2 archive_drives:2
  Catalog:
  mo.optic00          capacity:  1.2G space: 939.7M  -il-o--
-----
  mo.optic01          capacity:  1.2G space: 934.2M  -il-o--
-----
  mo.optic02          capacity:  1.2G space: 781.7M  -il-o--
-----
  mo.optic03          capacity:  1.2G space:   1.1G  -il-o--
-----
  mo.optic10          capacity:  1.2G space:  85.5M  -il-o--
-----
  mo.optic11          capacity:  1.2G space:  0      -il-o--
-----
  mo.optic12          capacity:  1.2G space: 618.9k  -il-o--
-----
  mo.optic13          capacity:  1.2G space: 981.3M  -il-o--
-----
  mo.optic20          capacity:  1.2G space:   1.1G  -il-o--
-----
  mo.optic21          capacity:  1.2G space:   1.1G  -il-o--
-----
  mo.optic22          capacity:  1.2G space: 244.9k  -il-o--
-----
  mo.optic23          capacity:  1.2G space:   1.1G  -il-o--
-----
```

The following output shows that both the metadata and data files are included in the archive set `samfs`. The archiver makes one copy of the files when their archive age reaches the default four minutes (240 seconds).

```
Archive file selections:
Filesystem samfs  Logfile:
samfs Metadata
    copy:1  arch_age:240
samfs1  path:.
    copy:1  arch_age:240
```

The following output shows the files in the archive sets archived to the volumes in the indicated order:

```
Archive sets:
allsets
samfs.1
 media: mo (by default)
 Volumes:
   optic00
   optic01
   optic02
   optic03
   optic10
   optic12
   optic13
   optic20
   optic21
   optic22
   optic23
 Total space available:   8.1G
```

Example 2

This example shows how to separate data files into two archive sets separate from the metadata. There is a manually mounted DLT tape drive in addition to the optical automated library from "Example 1" on page 124. The big files are archived to tape, and the small files are archived to optical cartridges.

Here is the `archiver.cmd` file for Example 2. The file is shown as output from the following command:

```
# archiver -lv -c example2.cmd
```

The following section of output is the content of the `archiver.cmd` file:

```
Reading archiver command file "example2.cmd"
1: # Example 2 archiver command file
2: # Simple selections based on size
3:
4: logfile = /var/opt/SUNWsamfs/archiver/log
5: interval = 5m
6:
7: # File selections.
8: big . -minsize 500k
9: all .
10:    1 30s
11:
12: vsns
13: samfs.1 mo .*0[0-2]      # Metadata to optic00 - optic02
14: all.1 mo .*0[3-9] .*[1-2][0-9]  # All others for files
15: big.1 lt .*
16: endvsns
```

Again, the media and drives to be used are shown, not the addition of the DLT and its defaults.Here is the organization of the file system. Files bigger than

```
Notify file: /opt/SUNWsamfs/sbin/archiver.sh
Archive media:
media:lt archmax: 512.0M Volume overflow not selected
media:mo archmax:   4.8M Volume overflow not selected
Archive libraries:
Device:hp30 drives_available:0 archive_drives:0
  Catalog:
  mo.optic00        capacity:  1.2G space: 939.7M  -il-o-----
--
  mo.optic01        capacity:  1.2G space: 934.2M  -il-o-----
--
  mo.optic02        capacity:  1.2G space: 781.7M  -il-o-----
--
  mo.optic03        capacity:  1.2G space:   1.1G  -il-o-----
--
  mo.optic04        capacity:  1.2G space: 983.2M  -il-o-----
--
  mo.optic10        capacity:  1.2G space:  85.5M  -il-o-----
--
  mo.optic11        capacity:  1.2G space:   0     -il-o-----
--
  mo.optic12        capacity:  1.2G space: 618.9k  -il-o-----
--
  mo.optic13        capacity:  1.2G space: 981.3M  -il-o-----
--
  mo.optic20        capacity:  1.2G space:   1.1G  -il-o-----
--
  mo.optic21        capacity:  1.2G space:   1.1G  -il-o-----
--
  mo.optic22        capacity:  1.2G space: 244.9k  -il-o-----
--
  mo.optic23        capacity:  1.2G space:   1.1G  -il-o-----
--
Device:lt40 drives_available:0 archive_drives:0
  Catalog:
  lt.TAPE01        capacity:  9.5G space:  8.5G  -il-o-------
  lt.TAPE02        capacity:  9.5G space:  6.2G  -il-o-------
  lt.TAPE03        capacity:  9.5G space:  3.6G  -il-o-------
  lt.TAPE04        capacity:  9.5G space:  8.5G  -il-o-------
  lt.TAPE05        capacity:  9.5G space:  8.5G  -il-o-------
  lt.TAPE06        capacity:  9.5G space:  7.4G  -il-o-------
```

512000 bytes (500 kilobytes) are archived after four minutes; all other files are archived after 30 seconds. Note the division of the archive sets among the

```
Archive file selections:
Filesystem samfs  Logfile: /var/opt/SUNWsamfs/archiver/log
samfs Metadata
    copy:1  arch_age:240
big  path:. minsize:502.0k
    copy:1  arch_age:240
all  path:.
    copy:1  arch_age:30
```

removable media in the following output:

```
Archive sets:
allsets
all.1
 media: mo
Volumes:
   optic03
   optic04
   optic10
   optic12
   optic13
   optic20
   optic21
   optic22
   optic23
 Total space available:   6.3G
big.1
 media: lt
Volumes:
   TAPE01
   TAPE02
   TAPE03
   TAPE04
   TAPE05
   TAPE06
 Total space available:  42.8G
samfs.1
 media: mo
Volumes:
   optic00
   optic01
   optic02
 Total space available:   2.6G
```

Example 3

In this example, user files and project data files are archived to various media. Files from the directory data are segregated by size to optical and tape media. Files assigned to the group ID `pict` are assigned to another set of volumes. Files in the directories `tmp` and `users/bob` are not archived. Archiving is performed on a 15-minute interval, and an archiving record is kept.

Figure 8. shows the output from the following command:

```
# archiver -lv -c example3.cmd
```

Figure 8. `archiver` **Command Output**

```
Reading archiver command file "example3.cmd"
1: # Example 3 archiver command file
2: # Segregation of users and data
3:
4: interval = 30s
5: logfile = /var/opt/SUNWsamfs/archiver/log
6:
7: no_archive tmp
8:
9: fs = samfs
10: no_archive users/bob
11: prod_big data -minsize 50k
12:    1 1m 30d
13:    2 3m
14: prod data
15:    1 1m
16: proj_1 projs/proj_1
17:    1 1m
18:    2 1m
19: joe . -user joe
20:    1 1m
21:    2 1m
22: pict . -group pict
23:    1 1m
24:    2 1m
25:
26: params
27: prod_big.1 -drives 2
28: prod_big.2 -drives 2
29: endparams
30:
31: vsns
32: samfs.1 mo optic0[0-1]$
33: joe.1 mo optic01$
34: pict.1 mo optic02$
35: pict.2 mo optic03$
```

Figure 8. `archiver` **Command Output (Continued)**

```
Reading archiver command file "example3.cmd”
36: proj_1.1 mo optic1[0-1]$
37: proj_1.2 mo optic1[2-3]$
38: prod.1 mo optic2.$
39: joe.2 lt 0[1-2]$
40: prod_big.1 lt 0[3-4]$
41: prod_big.2 lt 0[5-6]$
42: endvsns

Notify file: /opt/SUNWsamfs/sbin/archiver.sh

Archive media:
media:lt archmax: 512.0M Volume overflow not selected
media:mo archmax:   4.8M Volume overflow not selected

Archive libraries:
Device:hp30 drives_available:0 archive_drives:0
  Catalog:
  mo.optic00        capacity:  1.2G space: 939.7M  -il-
o-------
  mo.optic01        capacity:  1.2G space: 934.2M  -il-
o-------
  mo.optic02        capacity:  1.2G space: 781.7M  -il-
o-------
  mo.optic03        capacity:  1.2G space:   1.1G  -il-
o-------
  mo.optic04        capacity:  1.2G space: 983.2M  -il-
o-------
  mo.optic10        capacity:  1.2G space:  85.5M  -il-
o-------
  mo.optic11        capacity:  1.2G space:  0      -il-
o-------
  mo.optic12        capacity:  1.2G space: 618.9k  -il-
o-------
  mo.optic13        capacity:  1.2G space: 981.3M  -il-
o-------
  mo.optic20        capacity:  1.2G space:   1.1G  -il-
o-------
  mo.optic21        capacity:  1.2G space:   1.1G  -il-
o-------
  mo.optic22        capacity:  1.2G space: 244.9k  -il-
o-------
  mo.optic23        capacity:  1.2G space:   1.1G  -il-
o-------

Device:lt40 drives_available:0 archive_drives:0
  Catalog:
```

Figure 8. `archiver` **Command Output (Continued)**

```
Reading archiver command file "example3.cmd”
  lt.TAPE01          capacity:  9.5G space:  8.5G  -il-o----
---
  lt.TAPE02          capacity:  9.5G space:  6.2G  -il-o----
---
  lt.TAPE03          capacity:  9.5G space:  3.6G  -il-o----
---
  lt.TAPE04          capacity:  9.5G space:  8.5G  -il-o----
---
  lt.TAPE05          capacity:  9.5G space:  8.5G  -il-o----
---
  lt.TAPE06          capacity:  9.5G space:  7.4G  -il-o----
---

Archive file selections:
Filesystem samfs  Logfile: /var/opt/SUNWsamfs/archiver/log
samfs Metadata
    copy:1  arch_age:240
no_archive Noarchive path:users/bob
prod_big  path:data minsize:50.2k
    copy:1  arch_age:60 unarch_age:2592000
    copy:2  arch_age:180
prod  path:data
    copy:1  arch_age:60
proj_1  path:projs/proj_1
    copy:1  arch_age:60
    copy:2  arch_age:60
joe  path:. uid:10006
    copy:1  arch_age:60
    copy:2  arch_age:60
pict  path:. gid:8005
    copy:1  arch_age:60
    copy:2  arch_age:60
no_archive Noarchive path:tmp
samfs  path:.
    copy:1  arch_age:240

Archive sets:
allsets

joe.1
 media: mo
 Volumes:
   optic01
 Total space available: 934.2M

joe.2
 media: lt
 Volumes:
   TAPE01
   TAPE02
```

Figure 8. `archiver` **Command Output (Continued)**

```
Reading archiver command file "example3.cmd"
 Total space available:  14.7G

pict.1
 media: mo
 Volumes:
   optic02
 Total space available: 781.7M

pict.2
 media: mo
 Volumes:
   optic03
 Total space available:   1.1G

prod.1
 media: mo
 Volumes:
   optic20
   optic21
   optic22
   optic23
 Total space available:   3.3G

prod_big.1
 media: lt drives:2
 Volumes:
   TAPE03
   TAPE04
 Total space available:  12.1G

prod_big.2
 media: lt drives:2
 Volumes:
   TAPE05
   TAPE06
 Total space available:  16.0G

proj_1.1
 media: mo
 Volumes:
   optic10
 Total space available:  85.5M

proj_1.2
 media: mo
 Volumes:
   optic12
   optic13
 Total space available: 981.9M
```

Figure 8. `archiver` **Command Output (Continued)**

```
Reading archiver command file "example3.cmd”
samfs.1
 media: mo
 Volumes:
   optic00
   optic01
 Total space available:   1.8G
```

Example 4

In this example, user files and project data files are archived to the optical media. Note that Figure 9. does not use the directory structure pictured in Figure 7..

Four VSN pools are defined; three pools are used for user, data, and project, and one is a scratch pool. When the `proj_pool` runs out of media, it relies on the `scratch_pool` to reserve volumes. This example shows how to reserve volumes for each archive set based on the set component, owner component, and file system component. Archiving is performed on a 10-minute interval, and an archiving log is kept.

Figure 9. shows the `archiver.cmd` file and archiver output.

Figure 9. `archiver.cmd` **File and Archiver Output**

```
Reading archiver command file "example4.cmd"
1: # Example 4 archiver command file
2: # Using 4 VSN pools
3:
4: interval = 30s
5: logfile = /var/opt/SUNWsamfs/archiver/log
6:
7: fs = samfs
8: users users
9:     1 10m
10:
11: data data
12:    1 10m
13:
14: proj projects
15:    1 10m
16:
17: params
18: users.1 -reserve user
19: data.1 -reserve group
20: proj.1 -reserve dir -reserve fs
21: endparams
22:
23: vsnpools
24: users_pool mo optic0[1-3]$
```

Figure 9. `archiver.cmd` **File and Archiver Output (Continued)**

```
Reading archiver command file "example4.cmd"
25: data_pool mo optic1[0-1]$
26: proj_pool mo optic1[2-3]$
27: scratch_pool mo optic2.$
28: endvsnpools
29:
30: vsn
31: samfs.1 mo optic00
32: users.1 mo -pool users_pool -pool scratch_pool
33: data.1 mo -pool data_pool -pool scratch_pool
34: proj.1 mo -pool proj_pool -pool scratch_pool
35: endvsns

Notify file: /opt/SUNWsamfs/sbin/archiver.sh

Archive media:
media:mo archmax:   4.8M Volume overflow not selected

Archive libraries:
Device:hp30 drives_available:0 archive_drives:0
  Catalog:
  mo.optic00        capacity:  1.2G space: 939.7M  -il-o----
---
  mo.optic01        capacity:  1.2G space: 934.2M  -il-o----
---
  mo.optic02        capacity:  1.2G space: 781.7M  -il-o----
---
  mo.optic03        capacity:  1.2G space:   1.1G  -il-o----
---
  mo.optic04        capacity:  1.2G space: 983.2M  -il-o----
---
  mo.optic10        capacity:  1.2G space:  85.5M  -il-o----
---
  mo.optic11        capacity:  1.2G space:   0     -il-o----
---
  mo.optic12        capacity:  1.2G space: 618.9k  -il-o----
---
  mo.optic13        capacity:  1.2G space: 981.3M  -il-o----
---
  mo.optic20        capacity:  1.2G space:   1.1G  -il-o----
---
  mo.optic21        capacity:  1.2G space:   1.1G  -il-o----
---
  mo.optic22        capacity:  1.2G space: 244.9k  -il-o----
---
  mo.optic23        capacity:  1.2G space:   1.1G  -il-o----
---

Archive file selections:
Filesystem samfs  Logfile: /var/opt/SUNWsamfs/archiver/log
samfs Metadata
```

Figure 9. `archiver.cmd` **File and Archiver Output (Continued)**

```
Reading archiver command file "example4.cmd"
    copy:1  arch_age:240
users  path:users
    copy:1  arch_age:600
data  path:data
    copy:1  arch_age:600
proj  path:projects
    copy:1  arch_age:600
samfs  path:.
    copy:1  arch_age:240


VSN pools:
data_pool media: mo Volumes:
   optic10
 Total space available:  85.5M

proj_pool media: mo Volumes:
   optic12
   optic13
 Total space available: 981.9M

scratch_pool media: mo Volumes:
   optic20
   optic21
   optic22
   optic23
 Total space available:   3.3G

users_pool media: mo Volumes:
   optic01
   optic02
   optic03
 Total space available:   2.7G

Archive sets:
allsets

data.1
  reserve:/group/
 media: mo
 Volumes:
   optic10
   optic20
   optic21
   optic22
   optic23
 Total space available:   3.4G

proj.1
  reserve:/dir/fs
```

Figure 9. `archiver.cmd` **File and Archiver Output (Continued)**

```
Reading archiver command file "example4.cmd"
 media: mo
 Volumes:
   optic12
   optic13
   optic20
   optic21
   optic22
   optic23
 Total space available:   4.2G

samfs.1
 media: mo
 Volumes:
   optic00
 Total space available: 939.7M

users.1
  reserve:/user/
 media: mo
 Volumes:
   optic01
   optic02
   optic03
   optic20
   optic21
   optic22
   optic23
 Total space available:   6.0G
```

Archiver Guidelines

The archiver automates storage management operations using the `archiver.cmd` file. Before writing this file, it is useful to review some general guidelines that can improve the performance of your ASM or ASM-QFS file system and the archiver. This ensures that your data is stored in the safest way possible.

Each site is unique in its application of computing, data storage hardware, and software. The following recommendations are based upon the experiences of ASM. When writing the `archiver.cmd` file for your site, be sure that you reflect the data storage requirements at your site by considering the following aspects.

1. Save your archive logs. The archive logs provide information that is essential to recovering data, even when the ASM or ASM-QFS software is unavailable. It is recommended that you keep these logs in a safe place

in the event of a catastrophic disaster during which the ASM or ASM-QFS software is unavailable.

2. Use regular expressions for volumes. Let the system work for you by allowing it to put files on many different volumes. Volume ranges (specified using regular expressions) allow the system to run continuously. Using specific volume names for archive set copies can rapidly fill a volume, causing undue workflow problems as you remove a piece of media and replace it with another.
3. Base your archive interval on how often files are created and modified, and whether you want to save all modification copies saved. Remember, that the archive interval is the time between file system scans. A very short archive interval keeps the archiver scanning almost continuously.
4. Consider the number of file systems you are using. Multiple ASM and ASM-QFS file systems generally increase the performance of the archiver as compared to a single ASM or ASM-QFS file system. The archiver uses a separate process for each file system. Multiple file systems can be scanned in considerably less time than a single file system.
5. Use directory structures to organize your files within the ASM or ASM-QFS file system like UNIX file systems. For performance considerations, ASM recommends that you do not place more than 10,000 files in a directory.
6. Always make a minimum of two file copies on two separate volumes. Putting data on a single media type puts your data at risk if physical problems with the media occur. Do not rely on a single archive copy if at all possible.
7. Make sure you are dumping your metadata using `samfsdump`(1M) on a regular basis. The metadata (directory structure, file names, and so on) is stored in an archive set that has the same name as the file system. You can use this information to recover a file system in the event of a disaster. If you do not want to do this, you can prevent this data from being archived by assigning this archive set to a nonexistent VSN. For more information on preserving metadata, see the *ASM, ASM-QFS, and ASM/QFS-Standalone Disaster Recovery Guide* or the *ASM, ASM-QFS, and ASM/QFS-Standalone Installation and Configuration Guide*.

Troubleshooting the Archiver

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:

- `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
```

  Messages for each file system are displayed, including when the archiver will scan the `.inodes` file again and the files currently being archived.

- Archive logs. These logs are defined in the `archiver.cmd` file and should be monitored 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
```

The `archiver.sh` script creates two directories in `/var/opt/SUNWsamfs/archiver` to handle the exceptions of no space or no volumes for an archive set. These directories are named `NoSpace` and `NoVSNs`. These directories are populated with zero-sized files with the names of their respective archive sets. It is your responsibility to remove these files when the specific condition no longer exists. For more information, see the `archiver.sh`(4) man page.

Why Files Are Not Archiving

The following checklist includes reasons why your ASM or ASM-QFS environment might not be archiving files.

1. The `archiver.cmd` file has a syntax error. Run the `archiver -lv` command to identify the error, then correct the flagged lines.
2. The `archiver.cmd` file has a `wait` directive in it. Either remove the `wait` directive or override it from the `samu`(1M) utility's :`arrun` command.
3. No volumes are available. This is also shown with the `archiver -lv` command. Add more volumes as needed. You might have to export existing cartridges to free up slots in the automated library.
4. 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 177.
5. 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.
6. 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 ASM or ASM-QFS 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 archicing occurs. You should add cartridges, recycle, or use the `-sort path` parameter. For more information on these parameters, see “Associative Archiving” on page 109.
7. The `archiver.cmd` file has the `no_archive` directive set for directories or file systems that contain large files.
8. The `archive -n` (archive never) command has been used to set too many directories, and files are never archived.
9. Large files are busy. Thus, they never reach their archive age and are not archived.
10. Hardware or configuration problems exist with the automated library.
11. 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.

1. The syslog file (by default, `/var/adm/sam-log`). This file can contain archiver messages that can indicate the source of a problem.
2. Volume capacity. Ensure that all required volumes are available and have sufficient space on them for archiving.
3. 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.
4. You can use the `truss -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.
5. The `showqueue`(1M) command displays the content of the archiver queue files. This command can be used 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 “Troubleshooting the Releaser” on page 160.

Releasing

5

Releasing is the process by which the releaser makes disk cache space available by identifying archived files and releasing their disk cache copy. This makes room for other files to be created or staged from archive media. The releaser can release only archived files. Releasing the file results in a file without any data on the disk cache.

The ASM and ASM-QFS file systems invoke the releaser process when a site-specified disk threshold is reached. In contrast, the `release`(1) command allows users to immediately release a file's disk space or set releasing parameters for a file. For more information on the releaser process, see the `sam-releaser`(1M) man page.

The releaser contains features that allow you to specify that files be released immediately after archiving, that files never be released, or that files be partially released. The partial release feature is particularly useful because some applications, such as `filemgr`(1), read only the beginning of the file. With partial release, a portion of the file remains on the disk cache and the remainder of the file is released. Reading the first part of the file still on disk cache does not necessarily trigger the staging of the rest of the file back to disk cache from the archive media. These features, and many others, are described in this chapter.

This chapter contains the following topics:

Releaser Overview

When file system utilization exceeds its configured high watermark, the file system management software invokes the releaser. First, the releaser reads the `releaser.cmd` file and collects the directives that control the release process. Next, it scans the file system and collects information about each file. Finally, after scanning the entire file system, the releaser begins releasing files in priority order.

The releaser continues to release files as long as the file system remains above the configured low watermark. Typically, the releaser frees enough space to allow the file system to drop below the low water mark. If the releaser cannot find any files to release, it is forced to exit. The releaser runs later when more files can be released. While above the high watermark, the file system starts the releaser every one minute.

The high and low watermarks are set with the `high=`*percent* and `low=`*percent* file system mount options. For more information on these mount options, see the `mount_samfs`(1M) man page.

Theory of Operation

A file system can contain thousands of files. Keeping track of the release priority for all the files can be wasteful because releasing only several large files might return the file system to its low watermark. However, the releaser must examine the priority of each file or risk missing the best candidates for release. The releaser handles this condition by identifying only the first 10,000 candidates.

After identifying the first 10,000 candidates, the releaser discards subsequent candidates if they do not have a priority greater than the lowest-priority candidate among the first 10,000.

After the releaser has determined the priority of the first 10,000 candidates, it selects the files with the highest priority for release. After releasing each file, the releaser checks to see if the file system cache utilization is below the low watermark. If so, the releaser stops releasing files. If not, the releaser continues releasing the files in priority order.

If the releaser has released all 10,000 candidates and the file system is still above the low water mark, it starts over and identifies 10,000 new candidates.

The releaser exits if it cannot find any viable candidates. This can occur, for example, if files do not yet have archive copies. The ASM and ASM-QFS file systems start the releaser again after one minute has elapsed.

Definitions

Table 26. explains terms used throughout this chapter.

Table 26. Releaser Terms

Term	Definition
Age	The amount of elapsed time from a given event until now. A file's inode keeps track of the following times that are used by the releaser: residence-change time, data-modified time, and data-accessed time. You can view these times using the `sls`(1) command with the `-D` option. Each time has a corresponding age. For example, if it is 10:15 AM, a file with a modify time of 10:10 AM has a data-modified age of five minutes. For more information on the `sls`(1) command, see the `sls`(1) man page.
Candidate	A file that is eligible to be released. The reasons why a file would not be a candidate are as follows: • The file is already offline. • The file has not been archived. • The `archiver.cmd` command file specifies the `-norelease` attribute for the file, and the required copies have not yet been made. • The file is marked as damaged. • The file is not a regular file. It is a directory, block, character-special file, or pipe. • The archiver is staging the file to make an additional copy. The file becomes eligible for release after the archiver stages it. • The age of the file is negative. This usually occurs for NFS clients with inaccurate clock settings. • The file is marked `release -n`. • The file was staged at a time in the past that is less than the minimum residence time setting. For more information, see “Specifying a Minimum Residence Time” on page 156. • The file was flagged for partial release, by using the `release`(1) command’s `-p` option, and is already partially released. • The file is too small.

Table 26. Releaser Terms (Continued)

Term	Definition
Priority	Priority is a numeric value indicating the rank of a candidate file based on user-supplied weights that are applied to numeric attributes of that candidate. The overall priority is the sum of two types of priority: age priority and size priority. Candidate files with numerically larger priorities are released before candidates with numerically smaller priorities.
Weight	A numeric value that biases the priority calculation to include file attributes in which you are interested and to exclude file attributes in which you are not interested. For example, the size attribute of a file is excluded from the priority calculation if the size weight is set to zero. Weights are floating-point values from 0.0 to 1.0.
Partial release	A file can be partially released by specifying that a beginning portion of the file remain in disk cache while the rest of the file is released. For example, partial release is valuable when using utilities like `filemgr`(1) that read the beginning of a file.

■ Partial Release and Partial Stage

Releasing and staging are complementary processes. Files can be completely released from online disk cache after they are archived, or a site can specify that the beginning of a file (the *stub*) remain in disk cache while the remainder of the file is released. This ability to partially release a file provides immediate access to data in the file stub without staging the file.

A system administrator can specify both the default partial release size and the maximum size of the stub to remain online when a file system is mounted. The system administrator can set these on the `mount`(1M) command, as follows:

- The `-o partial=`*n* option sets the default size (*n*) of a file stub to remain online. The `-o partial=`*n* setting must be less than or equal to the `-o maxpartial=`*n* setting. The smallest possible setting is `-o partial=8` kilobytes. The default setting is `-o partial=16` kilobytes.
- The `-o maxpartial=`*n* option sets the maximum size (*n*) of a file stub to remain online. To limit the size of the file stub that can be left online, use the `-o maxpartial=`*n* option and specify a size equal to the largest stub that can be left online. To disable the partial release feature, specify `-o maxpartial=0`.

A user can specify the default stub size for a file by specifying the `-p` option on the `release`(1) command or the `p` option on the `sam_release`(3) library routine. To specify different-sized file stubs for different types of files or different applications, a user can specify the `-s` option on the `release`(1) command or the `s` option on the `sam_release`(3) library routine. The `-s` and `s` values must be less than the `-o maxpartial` value used on the `mount`(1M) command when the file system was mounted.

Another mount option, `-o partial_stage=`*n*, allows a system administrator to establish how much of a partial release stub must be read before the rest of the file is staged. That is, reading past the `-o partial_stage=`*n* size specification initiates the stage of the file.

By default, the `-o partial_stage=`*n* option is set to the size of the partial release stub. This value can be configured, though, and it affects file staging as follows:

- If the `-o partial_stage=`*n* option is set to the size of the partial release stub, the default behavior prevents the file from being staged until the application reaches the end of the partial release stub. Waiting until the end of the stub is reached causes a delay in accessing the rest of the file.
- If the `-o partial_stage=`*n* option is set to a value smaller than the partial release stub, the following occurs. After the application crosses the threshold set by the `-o partial_stage=`*n* option, the rest of the file is staged. This reduces the chance of a delay in accessing the rest of the file data.

Example. Assume that the following options are in effect:

- `-o partial_stage=16` (which is 16 kilobytes)
- `-o partial=2097152` (which is 2 gigabytes)
- `-o maxpartial=2097152` (which is 2 gigabytes)

The `filemgr`(1) program is being used, and it reads the first 8 kilobytes of a file. The file is not staged. A video-on-demand program reads the same file, and the file is staged after it reads past the first 16 kilobytes of the file. The application continues reading the 2 gigabytes of disk data while the archive tape is mounted and positioned. When the video-on-demand program reads past 2 gigabytes of file data, the application reads immediately behind the staging activity. The application does not wait because the tape mounting and positioning is done while the application reads the partial file data.

Several command line options affect whether a file can be marked for partial release. Some options are enabled by the system administrator, and others can be enabled by individual users. The following sections describe the release characteristics that can be set by the various types of users.

System Administrator Option Summary

The system administrator can change the maximum value and default value for partial release when the file system is mounted. The `mount`(1M) options in

Table 27. affect partial release. For more information on the `mount`(1) command, see the `mount_samfs`(1M) man page.

Table 27. Mount Options for Partial Release

`mount`**(1M) Option**	**Effect**
`-o maxpartial=`*n*	Determines the maximum amount of space, in kilobytes, that can remain in online disk cache if a file is marked for partial release. The maximum value is 2,097,152 kilobytes, which is 2 gigabytes. The minimum value is 0, which prevents any file from being partially released. If `-o maxpartial=0` is specified, the partial release feature is disabled, released files are released completely, and no portion of a file remains in disk cache. Users cannot override the value specified on this option after the file system is mounted. By default, the *n* argument is set to 16. This setting enables users to mark files for partial release with the maximum amount remaining on disk being 16 kilobytes.
`-o partial=`*n*	Sets a default amount of space, in kilobytes, to remain in disk cache if a user marks a file for partial release by using the `release`(1) command's `-p` option. The *n* argument must be at least 8, but it can be as great as the value specified for the `-o maxpartial=`*n* option. Because some applications do not need access to the entire file to complete their work, this option can be used to ensure that applications have the needed beginnings of files available to them. At the same time, using this option prevents files from being staged unnecessarily. The default value is `-o partial=16`.
`-o partial_stage=`*n*	Specifies that when a partially released file is accessed, *n* bytes of the file must be read before the entire file is staged from the archive media. This value is typically set to be lower than the amount of the `-o partial` setting. For *n*, specify an integer value from 0 to the `-o maxpartial` specification. By default, this is set to 16, or whatever value was specified for the `-o partial` option.
`-o stage_n_window=`*n*	Specifies the amount of data to be staged at any one time to *n*. For *n*, specify an integer from 64 to 2,048,000. The default is 256 kilobytes. This option applies only to files that have the `stage -n` attribute set.

User Option Summary

The system administrator sets maximum and default values for the size of a file stub that can remain in disk cache after the file is released. The system administrator also determines whether or not the partial release feature is enabled for a particular file system.

By using the `release`(1) command and the `sam_release`(3) library routines, however, a user can set other release attributes and can specify the files to be marked for partial release. The command and library options that determine partial release attributes are shown in Table 28.. For more information on the `release`(1) command, see the `release`(1) man page. For more information on the `sam_release`(3) library routine, see the `sam_release`(3) man page.

Table 28. User Release Options

Options	Effect
`release`(1) command and `-p` option or `sam_release`(3) library routine and `p` option	The `-p` and `p` options mark the named file for partial release. If these options are used, the amount of the file remaining in online disk cache after the file is released depends on the value of the `-o partial=`*n* option that was set when the file system in which the file resides was mounted. These options cannot be used to specify the number of bytes to remain online.
`release`(1) command and `-s partial_size` option or `sam_release`(3) library routine and `s` option	The `-s` and `s` options mark the named file for partial release, and they specify the amount of the file to remain in online disk cache. The arguments to the `-s` or `s` options specify the amount, in kilobytes, to remain online. A user cannot specify that the amount of a file remaining online be greater than the amount specified for the `-o maxpartial=`*n* value when the file system was mounted. If the user's value is greater than the value for the file system, the value for the file system is used, and the user's specification is ignored.

■ The releaser.cmd File

The /etc/opt/SUNWsamfs/releaser.cmd file consists of directive lines that specify site-specific releasing actions. The releaser.cmd file can contain directives for setting the release priority, specifying a log file, and other actions.

The following sections describe the releaser.cmd directives:

- "Specifying Age-Related and Size-Related Release Priority Directives" on page 152
- "Specifying Directives for Individual File Systems" on page 155
- "Specifying Debugging Directives" on page 155
- "Specifying a Minimum Residence Time" on page 156
- "Specifying a Log File" on page 156
- "Inhibiting Releasing for Rearchived Files" on page 158

For more information on these directives, see the releaser.cmd(4) man page.

Specifying Age-Related and Size-Related Release Priority Directives

Files are released from a file system using a priority order determined by directives defined in the releaser.cmd file. Both file age and file size are considered. By default, sites release the largest, oldest files first, leaving the smallest, newest files on disk. The following sections show how the releaser considers a file's age and size when determining the release priority of files in a file system.

For additional information on releaser directives, see the releaser.cmd(4) man page.

File Age

The releaser considers the following possible ages when determining the age-related component of a file's release priority:

- The age since it was last accessed
- The age since it was last modified
- The age since it changed residency in disk cache

In some cases, you might want the access age of a file to take precedence over the modification age. In other cases, a simple age derived from the most recently accessed time, modified time, and residence-changed time is preferred.

By default, the age of a file is the more recent of the file's three ages:

- File access age
- File modification age
- File residency age

You can use directives to specify that a weighted age priority be used when calculating the release priority for a file.

The formats of these age priority directives are as follows:

```
weight_age = float
weight_age_access = float
weight_age_modification = float
weight_age_residence = float
```

- The `weight_age` directive specifies that a file's default age (the smaller of the file's access, modification, or residence age) be given a weighting factor. For *float*, specify a floating-point number in the following range: 0.0 ≤ *float* ≤ 1.0. By default, *float* `= 1.0`.

  This directive cannot be specified in conjunction with the `weight_age_residence`, `weight_age_modify`, or `weight_age_access` directives.

- The `weight_age_residence`, `weight_age_modify`, and `weight_age_access` directives specify that that a file's age be determined based on a combination of one, two, or three of these possible ages. For *float*, specify a floating-point number in the following range: 0.0 ≤ *float* ≤ 1.0. By default, *float* `= 1.0`.

  These directives cannot be specified in conjunction with the `weight_age` directive.

  If the `weight_age_residence`, `weight_age_modify`, and `weight_age_access` directives are used, the age-related priority for a file is calculated based on a combination of all three ages. First, file age data is gathered for each file's possible age. Secondly, the file age data is multiplied by the weighting factors specified in the `releaser.cmd` file. Finally, the file's age-related priority is calculated by summing the product of the age data multiplied by each weighting factor, as shown in the following equation:

```
  file access age * weight_age_access
+ file modification age * weight_age_modification
+ file residency age * weight_age_residence

= age_related_priority
```

Example. The following lines in a `releaser.cmd` file specify that only the file's residence age be considered (and that the modification age and the access age be ignored) when calculating the release priority of a file:

```
weight_age_residence = 1.0
weight_age_modify = 0.0
weight_age_access = 0.0
```

After a file's age-related priority is calculated, it is multiplied by the file's size-related priority. The size-related priority is calculated as shown in the following section.

File Size

The releaser considers a file's size when determining the size-related component of a file's release priority. The size of the file (in 4-kilobyte blocks) is multiplied by the weight specified for the `weight_size` directive to obtain the size-related component of a file's release priority.

The format of the `weight_size` directive is as follows:

```
weight_size = float
```

For *float*, specify a floating-point number in the following range:
$0.0 \leq float \leq 1.0$. By default, *float* `= 1.0`.

Example. The following `releaser.cmd` file specifies that when calculating a file's release priority, a file's size is to be ignored for all files in the `samfs1` and `samfs2` file system.

```
# releaser.cmd file
logfile = /var/adm/default.releaser.log
weight_size = 0.0
#
fs = samfs1
weight_age = 1.0
logfile = /var/adm/samfs1.releaser.log
#
fs = samfs2
weight_age_modify = 0.3
weight_age_access = 0.03
weight_age_residence = 1.0
logfile = /var/adm/samfs2.releaser.log
```

Specifying Directives for Individual File Systems

You can use the `fs =` *family_set_name* directive in the `releaser.cmd` file to indicate that the directives that follow the `fs =` directive apply only to the named file system. This directive has the following format:

```
fs = family_set_name
```

For *family_set_name*, specify the name of a Family Set in the `mcf` file.

Directives preceding the first `fs =` directive are global and apply to all file systems. Directives following the `fs =` directive override global directives. The directives described in this chapter can be used as either global directive or as directives specific to one file system.

The `releaser.cmd`(4) man page includes examples of the `fs =` directive.

Specifying Debugging Directives

The `no_release` and `display_all_candidates` directives can be useful when tuning or debugging the releaser. These directives are as follows:

- The `no_release` directive prevents files from being removed from online disk cache. You can use this directive to check the directives in the `releaser.cmd` without actually releasing files. This directive has the following format:

```
no_release
```

- The `display_all_candidates` directive writes the names of all release candidates to the log file. This directive has the following format:

```
display_all_candidates
```

These directives are helpful when debugging because the releaser writes the names of release candidates to the log file, but it does not physically release them from the file system.

Specifying a Minimum Residence Time

The `min_residence_age` directive enables you to specify the minimum amount of time that a file must reside in a file system before it becomes a candidate for release. This directive has the following format:

```
min_residence_age = time
```

For *time*, specify a time in seconds. The default time is 600, which is 10 minutes. There is no practical minimum or maximum *time* setting.

Specifying a Log File

If a `logfile` directive is specified in the `releaser.cmd` file, the releaser either appends its activity log to the indicated file name, or the releaser creates the file name if it does not exist. This directive has the following format:

```
logfile = filename
```

For *filename*, specify the name of a log file.

The following is a sample log file (note that some lines have been wrapped to fit on the page):

```
Releaser begins at Wed Apr 28 17:29:06 1999
inode pathname          /sam1/.inodes
low-water mark          24%
weight_size             1
weight_age              1
fs equipment ordinal    1
family-set name         samfs1
started by sam-initd?    yes
release files?          yes
display_all_candidates? no
---before scan---
blocks_now_free:    3481504
lwm_blocks:         3729362
---scanning---
10501 (R: Wed Apr 21 18:47:50 CDT 1999) 10001 min, 500 blks /sam1/
testdir0/filevp
10500 (R: Wed Apr 21 18:48:10 CDT 1999) 10000 min, 500 blks /sam1/
testdir0/filewq
...
---after scan---
blocks_now_free:      3730736
lwm_blocks:           3729362
archnodrop: 0
already_offline: 0
bad_inode_number: 0
damaged: 0
extension_inode: 0
negative_age: 0
nodrop: 1
not_regular: 9
number_in_list: 675
released_files: 202
too_new_residence_time: 0
too_small: 2
total_candidates: 675
total_inodes: 1376
wrong_inode_number: 0
zero_arch_status: 689
zero_inode_number: 0
zero_mode: 0
CPU time: 2 seconds.
Elapsed time: 10 seconds.
Releaser ends at Wed Apr 28 17:29:16 1999
```

The `releaser`(1M) man page describes the information contained in the log file. Because the size of the log increases with each releaser run, be sure to allow for decreasing the size of the log, or omit the `logfile` keyword.

The following mathematical relationships exist among the statistics shown under the
`---after scan---` line:

```
total_inodes = wrong_inode_number +
zero_inode_number +
zero_mode +
not_regular +
extension_inode +
zero_arch_status +
already_offline +
damaged +
nodrop +
archnodrop +
too_new_residence_time +
too_small +
negative_age +
total_candidates
   released_files = total_candidates
```

Inhibiting Releasing for Rearchived Files

By default, files marked for rearchiving are released. If the `rearch_no_release` directive is specified in the `releaser.cmd`(4) fie, files marked for rearchiving are not released. This directive has the following format:

```
rearch_no_release
```

The archiver.cmd File's Role in Releasing

Most directives in the `archiver.cmd` file affect archiving, but the archive set assignment directive allows you to specify release attributes that apply to all files in an archive set.

The archive set assignment directive has the following format:

archive_set_name path [*search_criteria ...*] *directives* ...

Table 29. shows the *directives* that pertain to releasing.

Table 29. Archive Set Assignment *directives*

Directive	Effect
`-release a`	Specifies that the files in the archive set should be released after the first archive copy is made. Do not use this option if you are making more than one archive copy of each file. In such a situation, copy 1 would be be staged in order to make copy 2.
`-release n`	Specifies that the files in the archive set should never be released.
`-release p`	Specifies that the files in the archive set should be partially released after archiving.

For more information on these and the other `archiver.cmd` directives, see “Archiving” on page 77.

Configuring the Releaser

It is necessary to decide the characteristics of files in cache for your site. It is wasteful to load a tape if you are staging only a few kilobytes, so you may want to bias your system to retain small files in cache. To cause the releaser to release the largest files first, use the following directives in the `releaser.cmd` file:

```
weight_size = 1.0
weight_age = 0.0
```

Alternately, you may want to retain recently modified files in cache since a recently modified file might be modified again soon. This avoids the overhead created when the file is staged to enable modification. In this case, use the second set of age weights. To cause the releaser to weight files in strict order starting with the oldest modified to the most recently modified, use the following directives in the `releaser.cmd` file:

```
weight_size = 0.0
weight_age_access = 0.0
weight_age_modify = 1.0
weight_age_residence = 0.0
```

However, as the following examples demonstrate, most situations are not this straightforward.

Example 1. Assume that you want to release the largest files first. There are hundreds of small files that are the same size, and there are several large files. The cumulative size of the small files might exceed the size of the single, largest file. Eventually, the releaser releases all the large files. If `weight_age = 0.0` is specified, the releaser releases the small files in essentially random order because they are all the same size and have the same release priority.

In this scenario, you could set `weight_age = 0.01` as a tiebreaker. The releaser would release the older of two equally sized files first.

Example 2. This example presents a better method to specify how to release the largest files first.

Set `weight_size = 1.0` and `weight_age = 0.01`.

These directives violate the largest-first policy by counting smaller, less recently accessed files as better candidates than larger, more recently accessed files. You can make this effect as small as you want by making `weight_age` smaller than `weight_size`. For example, based on the previous settings, a 4-kilobyte file that staged 100 minutes ago and an 8-kilobyte file that just staged both have the same release priority.

The releaser randomly chooses a file to release. If it chooses a 4-kilobyte file, it violates the largest-first intent. Setting `weight_age` considerably smaller (for example, to 0.001) reduces this effect. If a 4-kilobyte file staged 1,000 minutes ago, it has the same priority as the 8-kilobyte file that just staged.

You can use the `no_release` and `display_all_candidates` directives and run the releaser manually to obtain a list of candidates in priority order for use in adjusting the priority weights.

Running the Releaser Manually

From time to time, you might want to run the releaser manually. For this, you need to know the mount point of the file system and the low watermark the releaser should attempt to reach.

For example, to release files in the `/sam1` file system until it reaches 47 percent full. Log in as root and type the following:

```
# /opt/SUNWsamfs/sbin/sam-releaser /sam1 47 1.0
```

The final argument, `weight-size`, is overridden by a `weight_size` command in the `releaser.cmd` file.

As the releaser runs, it writes information to your screen and to the releaser log file (if specified in the `releaser.cmd` file.)

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 on this, see “Why Files Are Not Archiving” on page 142.
- 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 `libmgr`(1M), 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 `libmgr`(1M), and lower it.
- Large files are busy. They will never reach their archive age, never be archived, and never be released.

Staging

6

Staging is the process of copying file data from nearline or offline storage back to online storage. The staging capabilities enable you to stage files immediately, to never stage files, to specify partial staging, and to specify other staging actions. The never-stage capability can be used, for example, by applications that randomly access small records from large files; when this is enabled, the data is accessed directly from the archive media without staging the file online.

This chapter describes the ASM and ASM-QFS file staging capability. It contains the following topics:

The stager.cmd File

You can use the `stager.cmd` file to specify the stager's behavior. The full path name to this file is `/etc/opt/SUNWsamfs/stager.cmd`. By default, the stager performs the following actions:

- The stager attempts to use all the drives in the library to stage files.
- The stage buffer size is determined by the media type, and the stage buffer is not locked.
- No log file is written.
- Up to 1000 stage requests can be active at any one time.

The `stager.cmd` file allows you to specify directives to override the default behaviors. The rest of this section describes the stager directives. For additional information on stager directives, see the `stager.cmd`(4) man page.

The "Example stager.cmd File" on page 168 shows the completed `stager.cmd` file after all possible directives have been set.

The examples in this section assume the configuration defined in the following `mcf` file:

```
#
# Sun SAM-FS 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
#
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 dog  on /var/opt/SUNWsamfs/catalog/
dogcat
/dev/samst/c1t5u0 31 od dog    on
/dev/samst/c1t6u0 32 od dog    on
/dev/rmt/0cbn     40 od -      on
/dev/samst/c1t3u1 50 rb bird on /var/opt/SUNWsamfs/catalog/
birdcat
/dev/rmt/2cbn     51 tp bird   on
```

Specifying the Number of Drives

By default, the stager uses all available drives when staging files. If the stager keeps all the drives busy, this can interfere with the archiver's activities. The `drives` directive specifies the number of drives available to the stager. This directive has the following format:

```
drives = library count
```

where

library The Family Set name of a library as it appears in the ASM or ASM-QFS `mcf` file.

count The maximum number of drives to be used. By default, this is the number of drives configured in the `mcf` file for this library.

For example, the following directive line specifies that only one drive from the `dog` family set's library be used for staging files:

```
drives = dog 1
```

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

Setting the Stage Buffer Size

By default, a file being staged is read into memory in a buffer prior to restoring the file from the archive media back to online disk cache. You can use the `bufsize` directive to specify a nondefault buffer size and, optionally, to lock the buffer. These actions can improve performance, and you can experiment with various *buffer_size* values. This directive has the following format:

```
bufsize = media buffer_size [ lock ]
```

where:

media	Specify the archive media type from the list on the `mcf`(4) man page.
buffer_size	Specify a number from 2 through 32. The default is 4. This value is multiplied by the *dev*`_blksize` value for the media type, and the resulting buffer size is used. The *dev*`_blksize` can be specified in the `defaults.conf` file. The higher the number specified for *buffer_size*, the more memory is used. For more information on this file, see the `defaults.conf`(4) man page.
`lock`	The `lock` argument indicates whether or not the archiver should use locked buffers when staging archive copies. If `lock` is specified, the archiver sets file locks on the archive buffer in memory for the duration of the copy operation. This avoids the overhead of locking and unlocking the buffer for each I/O request and can result in a reduction in system CPU time. The `lock` argument should be specified only on large systems with large amounts of memory. Insufficient memory can cause an out-of-memory condition. The `lock` argument is effective only if direct I/O is enabled for the file being staged. By default, `lock` is not specified and the file system sets the locks on all direct I/O buffers, including those for archiving. For more information on enabling direct I/O, see the `setfa`(1) man page, the `sam_setfa`(3) library routine man page, or the `-O forcedirectio` option on the `mount_samfs`(1M) man page.

For example, this directive can be specified in the `stager.cmd` file in a line such as the following:

```
bufsize=od 8 lock
```

Specifying a Log File

You can request that the ASM or ASM-QFS file system collect file-staging event information and write it to a log file. The `logfile` directive specifies a log file to which the stager can write logging information. This directive has the following format:

```
logfile=filename
```

For *filename*, specify a full path name.

When a log file is specified, the stager writes a line to the log file for each file staged. This line includes information such as the name of the file, the date and time of the stage, and the VSN. For example, the following directive line specifies file `/var/adm/stage.log`:

```
logfile=/var/adm/stage.log
```

Figure 10. shows an example of a stager log file.

Figure 10. Stager Log File Example

```
S 10/24 09:30:51 mo opt02b 29405.fa7d 24.47 4699763 /sam1/
testdir0/filec
S 10/24 09:30:52 mo opt02b 29405.11e5a 25.47 1452980 /sam1/
testdir0/fileb
S 10/24 09:30:58 mo opt02b 29405.12971 26.47 4194084 /sam1/
testdir0/filea
S 10/24 09:31:04 mo opt02b 29405.2 13.47 4121178 /sam1/
testdir0/filel
S 10/24 09:31:08 mo opt02b 29405.1f75 14.47 2532411 /sam1/
testdir0/filek
S 10/24 09:31:11 mo opt02b 29405.32c9 15.47 2919620 /sam1/
testdir0/filej
S 10/24 09:31:16 mo opt02b 29405.4911 16.47 4173923 /sam1/
testdir0/filei
S 10/24 09:31:21 mo opt02b 29405.68eb 17.47 4714187 /sam1/
testdir0/fileh
S 10/24 09:31:24 mo opt02b 29405.8ce4 19.47 2595485 /sam1/
testdir0/fileg
S 10/24 09:31:28 mo opt02b 29405.a0b3 20.47 3952040 /sam1/
testdir0/filef
S 10/24 09:31:33 mo opt02b 29405.bedb 21.47 4344648 /sam1/
testdir0/filee
S 10/24 09:31:37 mo opt02b 29405.e002 22.47 3470154 /sam1/
testdir0/filed
```

As Figure 10. shows, the stager log file consists of lines of information divided into nine fields. Table 30. describes the content of the stager log file fields.

Table 30. Stager Log File Fields

Field	Content Description
1	Stage activity. `S` for staged. `C` for canceled. `E` for error.
2	Date of stage action in *mm*/*dd* format.
3	Time of stage action in *hh*:*mm*:*ss* format.

Table 30. Stager Log File Fields (Continued)

Field	Content Description
4	Archive media type. For information on media types, see the `mcf`(4) man page.
5	VSN.
6	Physical position of start of archive file on media (`tar`(1) file) and file offset on the archive file in hexadecimal.
7	Inode number and generation number. The generation number is an additional number used in addition ot the inode number for uniqueness since inode numbers get re-used.
8	Length of file.
9	Name of file.

Specifying the Number of Stage Requests

You can specify the number of stage requests that can be active at any one time by using the `maxactive` directive. This directive has the following format:

```
maxactive=number
```

By default, *number* is 1000. The minimum number allowed is 1.

For example, the following directive line specifies that no more than 500 stage requests can be in the queue simultaneously:

```
maxactive=500
```

Example stager.cmd File

The following is an example `stager.cmd` file:

```
# This is stager.cmd file /etc/opt/SUNWsamfs/stager.cmd
drives=dog 1
bufsize=od 8 lock
logfile=/var/adm/stage.log
maxactive=500
```

■ The archiver.cmd File’s Role in Staging

Most directives in the `archiver.cmd` file affect archiving, but the archive set assignment directive allows you to specify stage attributes that apply to all files in an archive set. The archive set assignment directive has the following format:

```
archive_set_name path [search_criteria ...] directives ... ]
```

Table 31. shows the *directives* that pertain to staging.

Table 31. Staging *directives*

directive	**Effect**
`-stage a`	Specifies that the files in the archive set should be associatively staged.
`-stage n`	Specifies that the files in the archive set should never be staged.

For more information on these and the other `archiver.cmd` directives, see "Archiving" on page 77.

Prioritizing Preview Requests Using the preview.cmd File

The archiver and stager processes both can request that media be loaded and unloaded. If the number of requests exceeds the number of drives available for media loads, the excess number of requests is sent to the preview queue.

Archive and stage requests in the preview queue are those that cannot be immediately satisfied. By default, preview requests are satisfied in first-in-first-out (FIFO) order.

The number of entries that can be in the preview queue is determined by the `previews=` directive in the `defaults.conf` file. For information on changing the value of this directive, see the `defaults.conf`(4) man page.

You can assign different priorities to preview requests. You can override the FIFO default by entering directives in the preview command file, which is written to the following location:

```
/etc/opt/SUNWsamfs/preview.cmd
```

This file schedules preview requests based on whether the request is for file staging or archiving. You can also increase the priority for specific VSNs. Further, settings in the `preview.cmd` file can also reprioritize preview requests for all or for specific file systems based on the high watermark (HWM) or low watermark (LWM) settings.

The preview directives are read by the `sam-initd` daemon at startup. The directives must be listed one per line. Changes made to this file while the `sam-initd` daemon is running do not take effect until the `sam-initd` daemon is restarted. Comment lines begin with a pound sign (#) and extend through the end of the line. For more information on this file, see the `preview.cmd`(4) man page.

The following two types of directives can appear in the `preview.cmd` file:

- Global directives, which apply to all file systems. These must appear before the first `fs =` line.
- Directive that are specific to a file system, which follow the global directives. Like the `archiver.cmd` file, the `preview.cmd` file can contain directives specific to individual file systems. The directives specific to individual file systems must appear in the file after all global directives.

  The file system directives must begin with an `fs =` *file_system_name*. directive. This directive names the file system to which all subsequent directives pertain. More than one block of file directives can appear in a file. File system directives apply until the next `fs =` line is encountered or until the end of file is encountered.

Note: When multiple directives affect a file system, the directives that are specific to a particular file system override the global directives.

VSN and Age Directives (Global)

The VSN and age priority directives are global directives. If they are present in your `preview.cmd` file, they must appear before any directives that are specific to a file system. That is, they must appear prior to any `fs =` directives. The VSN priority directive has the following format:

```
vsn_priority = value
```

This directive is a static priority factor. It indicates the value by which the total priority increases for a VSN flagged as a high-priority VSN. The default value for `vsn_priority` is 1000.0. VSNs must have their priority flag set when they are scheduled as preview requests to gain this value. Use the `chmed`(1M) command to set the priority flag with the `p` option (for example, `chmed +p lt.AAA123`). Setting this flag takes effect for all submitted requests for the VSN that are not already preview requests. The age priority directive has the following format:

```
age_priority = factor
```

This directive is a static priority factor. Its overall effect is dynamic. The `age_priority` factor is multiplied by the number of seconds a request is a preview request. The result is added to the overall priority of the request. The longer a request waits to be satisfied, the larger the age factor becomes.

Setting this factor helps to ensure that older requests are not indefinitely superseded by newer requests with other higher-priority factors.

If this factor is more than 1.0, it increases the importance of the time factor in calculating the total priority. If it is less than 1.0, it decreases the importance of the time factor. Setting the factor to 0.0 eliminates the time factor from the overall priority calculation.

A VSN whose priority flag is not set increases in priority based on the time it remains in the queue. Its priority can become higher than a VSN that comes into the queue later with the priority flag already set.

Watermark Directives (Global or File System Specific)

The watermark preview request directives can be used as either global or file system specific directives. The watermark priority directives determine the watermark priority (`wm_priority`) of the preview requests. The `wm_priority` factor is the sum of the following settings:

```
wm_priority = lwm_priority + lhwm_priority +
hlwm_priority + hwm_priority
```

When the `wm_priority` factor is a positive number, the result on the overall calculated priorities increases archiving requests over staging requests. However, the `wm_priority` factor can also be a negative number. In this case, the overall priority for archiving requests is reduced, which tends to favor staging requests over archival requests. A setting of 0.0 (or no specified command at all) indicates that no special action occurs to archival requests when the file system is in this condition. For more information on this, see the example in “Example 1: Enforcing Stage Requests” on page 174.

Table 32. shows the four watermark priority directives and their arguments

Table 32. Watermark Priority Directives

Priority Directive	Argument
`lwm_priority =` *value*	For *value*, specify the amount by which the `wm_priority` factor changes for archiving requests when the file system is below the LWM level. The default is 0.0.
`lhwm_priority =` *value*	For *value*, specify the amount by which the `wm_priority` factor changes for archiving requests when the file system crosses from below to above the LWM but remains below the HWM level. This generally indicates that the file system is filling up. The default is 0.0.

Table 32. Watermark Priority Directives

Priority Directive	Argument
`hlwm_priority =` *value*	For *value*, specify the amount by which the `wm_priority` factor changes for archiving requests when the file system has crossed from above to below the HWM but remains above the LWM level. This generally indicates that the releaser was not able to free enough disk space to leave the file system below LWM. The default is 0.0.
`hwm_priority =` *value*	For *value*, specify the amount by which the `wm_priority` factor changes for archiving requests when the file system is above the HWM level. The default is 0.0.

Together, the four watermark settings create a dynamic priority factor that includes a percentage value indicating how full the file system is and the levels at which the HWM and LWM are set. The value assigned to a preview request is determined by whether a factor is global, specific to a file system, or not set.

When a file system crosses from one condition to another, the priority of each VSN associated with that file system is recalculated based on the appropriate watermark priority setting, with or without the `chmed`(1M) command's `p` option.

The watermark priorities are used only to calculate media requests for archiving. They are not used to calculate media requests for staging.

The following example directives show how to slightly increase the priority for archiving requests when the file system is at HLWM. These example settings allow the releaser to free enough disk space so that the file system gets below LWM.

```
lhwm_priority = -200.0
hlwm_priority = 100.0
```

Calculating Total Preview Request Priority

The numeric priority of preview requests is determined by combining several static and dynamic factors. Higher numbers correspond to higher priority. A static priority factor is set when the request is generated. Its effect does not change the overall priority after the request is generated and is waiting to be satisfied. A dynamic priority factor can increase or decrease the overall priority of a request while the request is waiting to be satisfied.

The total priority for a preview request is the sum of all priority factors. It is calculated as follows:

```
  vsn_priority
+ wm_priority
+ (age_priority * time_in_sec_as_preview_request)
= priority
```

How to Set Up a Preview Request Priority Scheme

It is necessary to change the default preview request FIFO scheme only when there are compelling system reasons to do so. The following possible conditions might necessitate changing the default preview request FIFO scheme:

- Condition 1: Ensure that staging requests are processed before archive requests.
- Condition 2: Ensure that archive requests gain top priority when a file system is about to fill up.
- Condition 3: Push requests that use a specific group of media to the top of the preview request list.

For environments in which user access to data is of paramount importance, the VSN drives are limited, or file archival is performed as a background function, you can use the `preview.cmd` file to influence how the storage system resources service the staging requests. You can customize the settings in the `preview.cmd` file to support any of the preceding scenarios and influence the configured ASM or ASM-QFS environment.

Because data is not affected by the settings in this file, you are encouraged to experiment and adjust the directive settings to achieve the proper balance between archiving and staging requests when weighed against the priorities of each preview request.

The following example `preview.cmd` file addresses the three conditions listed previously:

```
# condition 1
lwm_priority = -200.0
lhwm_priority = -200.0
hlwm_priority = -200.0
# condition 2
hwm_priority = 500.0
# condition 3
age_priority = 1.0
```

Example 1: Enforcing Stage Requests

The following example settings demonstrate one way to ensure that stage requests have priority over archive requests. This example assumes the following:

- Several requests are sitting in the queue for 100 seconds.
- The default `vsn_priority` is 1000.

Table 33. shows how the total request priorities are calculated.

Table 33. Request Priority Example

Priority	Calculation
Archive VSN with priority, LWM:	1000 + (-200) + (1 x 100) = 900
Stage VSN with priority, LWM:	1000 + 0 + (1 x 100) = 1100
Stage VSN without priority, LWM:	0 + 0 + (1 x 100) = 100

This example shows that a negative value for `wm_priority` tends to favor staging requests over archival requests when the other factors are equal.

Example 2: Enforcing Archive Requests

When the environment is balanced between the importance of staging a file back to the user versus getting new files archived to media, the biggest concern is exceeding the HWM. In this situation, if there are not enough files who have met their archive requirements to lower the percent full of the file system, completing the pending archive requests is the next best way to keep the file system from filling up.

In this situation, the `preview.cmd` file can be as simple as the following:

```
hwm_priority = 500.0
```

Example 3: Prioritizing Requests By Media

In project-oriented environments, specific users might be working on groups of files that use specific VSNs and are segregated from other users. In this environment, certain projects might have higher priorities at certain times; hence, greater priority might be required from the available system storage resources. You can configure the `preview.cmd` file with the following directive to give users and their media the appropriate priority for media drives:

```
hwm_priority = 5000.0
```

Then, for every VSN in the priority user's group, enter the following information:

```
chmed +p lt.AAA123 ## or whatever VSN is used
```

Thereafter, every request that requires VSN `AAA123` (or whatever VSN is used) is placed above other pending mount requests in the preview queue.

Later, to deprioritize the user's media, do the following reverse command for every VSN:

```
chmed -p lt.AAA123 ## or whatever media type is used
```

Example 4: Complex Prioritization

Assume that there are two ASM file systems with the following requirements:

- No request should sit too long in the queue (`age_priority`).
- When a file system is below the LWM, staging requests should take precedence.
- When a file system is above the LWM but below the HWM, it is not necessary to prioritize archive or stage requests one over the other. In this case, the affected directives are as follows:

```
lwm_priority =  -200.0
lhwm_priority = 0.0
hlwm_priority = 0.0
```

In this case, the other directives remain unchanged.

When a file system goes over the HWM, archive requests should take priority.

If both file systems are over the HWM, it is more important to prevent the second file system (for example, `samfs2`) from filling up. This might occur if, for example, `samfs1` is a user working file system and `samfs2` is the critical-system file system.

In all cases, regardless of the situation, a request for a select group of VSNs takes precedence in the preview request queue if the `chmed`(1M) command's `p` flag is set.

The following `preview.cmd` file prioritizes requests according to the requirements in the preceding list:

```
age_priority = 100.0
vsn_priority = 20000.0
lhwm_priority = -200.0
hlwm_priority = -200.0
fs = samfs1
hwm_priority = 1000.0
fs = samfs2
hwm_priority = 5000.0
```

Recycling

7

Recycling is the process of reclaiming space on archive volumes. The recycler works with the archiver to reclaim the space occupied by unused archive copies. As users modify files, the archive copies associated with the old versions can be purged from the system. The recycler identifies the volumes with the largest proportions of expired archive copies and directs the moving of unexpired copies to different volumes. If only expired copies exist on a given volume, a site-defined action is taken. For example, such a volume can be relabeled for immediate reuse or exported to offsite storage, thus keeping a separate historical record of file changes. Users are unaware of the recycling process as it relates to their data files.

This chapter includes the following topics:

Overview

The recycler is responsible for keeping the amount of space consumed by expired archive copies to a minimum as defined by site-specified parameters. At any time, the space on a given archive volume consists of the following:

- *Current data* is space being used for archive images that are active currently.
- *Expired data* is space used by archive images that are no longer active currently.
- *Free space* is space that is not being used by currently active or expired archive images.

The *capacity* of a volume is the total amount of space for data on a volume. For example, a 10-gigabyte tape volume with 3 gigabytes written to it has a capacity of 10 gigabytes and 7 gigabytes of free space.

New or newly labeled archive media starts out with all its capacity as free space. As data is archived to the media, the amount of free space decreases and the amount of current data increases.

As archived files in the file system are changed or removed, their archive images expire and they move from the current data classification to the expired data classification. The physical space used by these images remains the same; there is simply no longer a file in the file system pointing to that space.

These expired images (and thus, expired data) would eventually consume all free space. Only when space is recycled can these images be removed and the space they occupy become free. The goal of the recycler is to transform space used by expired data into free space without losing any current data.

Tape cartridges can be only appended to. They cannot be rewritten in place. The only way to reuse tape cartridges is to move all the current data off a cartridge, relabel the cartridge, and start using it again from the beginning. To achieve this, the archiver identifies all the current archive images present on a volume. It marks these images to enable the archiver to replace the copy on the volume being recycled with a copy on another volume. This operation is called *rearchiving*. You can use the `sls`(1) command and its `-D` option to display information about a file, and the output from the `sls`(1) command shows whether or not a file is scheduled for rearchiving.

The recycler never actually moves files to new media. For all files on the selected volume, it sets the rearchive file attribute, and the archiver picks it up later. The recycler sets the `recycle` attribute on the selected media, so it receives no new data during archiving. The archiver does the rest of the work. The archiver acts on files with the `rearchive` attribute by actually moving their archive copies off of the subject media and onto the new.

After all the archive images on the VSN have been rearchived, the VSN contains only free space and expired space. At that time, it is safe to relabel the cartridge.

Recycling is initiated by entering the `sam-recycler`(1M) command. This can be done manually or through a `cron`(1) job. You can recycle in one of the following ways:

- By automated library utilization
- By archive set utilization

Recycling progresses a little differently depending on the archive media. The following sections describe the recycling process for recycling disk archive copies and for recycling removable media volumes.

Recycling Disk Archive Copies

Recycling expired archive copies that have been written to disk is a simpler process than recycling removable media volumes. When you recycle disk volumes, files are never rearchived to other media. Expired archive copies are removed from the disk because there is no need to rearchive.

The only way to recycle disk archive copies is to recycle by archive set. You cannot recycle by library. No `recycler.cmd` file is necessary. All recycling activities for disk archive copies are controlled by directives placed in the `archiver.cmd` file.

Recycling Removable Media Archive Copies

The recycler is designed to run periodically. It performs as much work as it can each time it is invoked. Between executions, the recycler keeps state information in the library catalogs and the inodes.

When the recycler is run, it finishes its work long before the data is actually moved to new media. In fact, the recycler must finish in order for rearchiving to be successful. If the archiver does not run thereafter, or if media is not available, or if any other archiver anomaly arises, files with the `rearchive` attribute are not rearchived to new media. In this case, the old media is never drained. Furthermore, if the archiver does not rearchive all the files, when the recycler runs the next time (looking for media to relabel - drained as a result of the previous recycler and archiver runs combined), the media cannot be relabeled and reused because the media has not been drained of valid archive copies.

■ Recycling Directives

The `recycler.cmd` file accepts the directives described in the following sections:

- "The logfile Directive" on page 179
- "The no_recycle Directive" on page 180
- "The Library Directive" on page 180

The logfile Directive

The `logfile` directive specifies a recycler log file. This directive has the following format:

```
logfile = filename
```

where:

filename	Specify a path to the log file.

The following is an example of a `logfile=` directive line:

```
logfile=/var/adm/recycler.log
```

The no_recycle Directive

The `no_recycle` directive enables you to prevent recycling of volumes. To specify the VSNs, you use regular expressions and one or more specific media types. This directive has the following format:

```
no_recycle media_type VSN_regex [ VSN_regex ... ]
```

where:

media_type	Specify a media type from the `mcf`(4) man page.
VSN_regexp	Specify one or more space-separated regular expressions to describe the volumes. For information on the format of a regex, see the `regexp`(5) man page or see “File Name search_criteria Using Pattern Matching: -name regex” on page 99.

By specifying a *media_type*, you can prevent the recycling of volumes stored on a particular type of media. One or more *VSN_regexp* specifications enables you to use a regular expression to identify specific cartridges to be excluded from recycling.

For example, the following directive line excludes from recycling any tape volumes whose VSN identifiers begin with `DLT`:

```
no_recycle lt DLT.*
```

The Library Directive

The library directive enables you to specify various recycling parameters for the VSNs associated with a specific library. This directive has the following format:

```
library parameter [ parameter ... ]
```

where:

library	Specify the library's name as specified in the Family Set field of the mcf(4) file.
parameter	Specify one or more space-separated *parameter* keywords from Table 34..

Table 34. Library Directive *parameter* Values

parameter	Action
`-dataquantity` *size*	Limits the amount of data that the recycler can schedule for rearchiving in its efforts to clear volumes of useful data. Default is 1 gigabyte.
`-hwm` *percent*	Library high watermark. Default is 95.
`-ignore`	Prevents volumes in this library from being recycled. This directive is useful when testing the `recycler.cmd` file.
`-mail [` *email_address* `]`	Sends email messages to the designated *email_address*. By default, no email is sent. If `-mail` is specified with no argument, email is sent to `root`.
`-mingain` *value*	Minimum VSN gain. Default is 50.
`-vsncount` *count*	Limits the number of volumes to be recycled to count. Default is 1.

For example, consider the following directive line:

```
gr47 -hwm 85 -ignore -mail root -mingain 40
```

It specifies the following for library `gr47`:

- The library should be considered for recycling when the volumes in the library are 85 percent full.
- The minimum percent gain is 40 percent.
- No more than 1 gigabyte is to be rearchived. This is the default, so it is not specified in the `recycler.cmd` file.
- Only one volume is to be recycled. This is also a default setting.
- Recycling messages are emailed to `root`.

Configuring the Recycler

Prior to configuring the recycler, note the following:

- Directives in the `archiver.cmd` file control recycling by archive set. Directives in the `recycler.cmd` file control recycling by library. In addition, the `recycler.cmd` file controls general recycler behavior. For information on recycler directives, see “Recycling Directives” on page 179.
- The recycler should not be used on a volume that contains any removable media files. Removable media files are created using the `request`(1) command. The recycler does not preserve removable media files created by the `request`(1) command. A volume with removable media files can never be drained.
- Do not run the recycler while performing maintenance on a ASM or ASM-QFS file system. The recycler uses the `.inodes` file and the `mcf` file to help identify files that are current or expired and the devices associated with a file system. Absence of proper information in these files can cause current archived data to appear as expired and be recycled.
- All ASM and ASM-QFS file systems must be mounted when the recycler is run. If you are recycling from online disk, the file system that contains the disk volumes must be mounted and the host system must be accessible.

The recycler is not enabled by default. You must initiate recycling by entering the `sam-recycler`(1M) command. When the recycler is initiated, the default recycler settings specified in “The Library Directive” on page 180 take effect. For more information on the recycler, see the `sam-recycler`(1M) man page.

The following sections describe the process for configuring the recycler. This process includes several steps, and the steps are as follows:

- “Step 1: Creating a recycler.cmd File (Optional)” on page 182
- “Step 2: Editing the archiver.cmd File (Optional)” on page 185
- “Step 3: Running the Recycler” on page 186
- “Step 4: Creating a crontab File for the Recycler (Optional)” on page 188
- “Step 5: Removing -recycle_ignore and ignore Directives” on page 188
- “Step 6: Creating a recycler.sh File” on page 188

If you are recycling to cartridges in a library, this process includes creating a `recycler.cmd` file and, optionally, editing the `archiver.cmd` file. If you are archiving to disk, you can archive only by archive set, so to enable recycling of these disk volumes, you edit the `archiver.cmd` file. The following procedure describes configuring the recycler for any archive media.

Step 1: Creating a recycler.cmd File (Optional)

Perform this step if you are recycling archive copies on cartridges in a library.

If you are recycling archive copies on disk volumes, you cannot complete this step because recycling is controlled by directives in the `archiver.cmd` file. For information on the configuring recycling in the `archiver.cmd` file, see “Step 2: Editing the archiver.cmd File (Optional)” on page 185.

The `recycler.cmd` file contains general recycling directives and can also contain directives for each library in the ASM or ASM-QFS environment. For information on the recycling directive, see “Recycling Directives” on page 179.

Even if you are recycling by archive set, you still should configure each library in the `recycler.cmd` file. This ensures that VSNs that do not fall into an archive set can be recycled if needed.

A typical `recycler.cmd` file contains the following directive lines:

- A `logfile=` directive line to specify a recycler log file. The system writes recycling messages and recycling reports to this file.
- One or more directive lines for each library that contains volumes to be recycled. This line must contain the family set name (from the `mcf` file) for the library being recycled. This identifies the library to the recycler.

Because you are still creating the `recycler.cmd` line, and it has not yet been tested, the `ignore` keyword should be used. The `ignore` keyword is removed in a later step in this process.

Example recycler.cmd File

Figure 11. shows an example of a `recycler.cmd` file.

Figure 11. A `recycler.cmd` File Example

```
logfile = /usr/tmp/recycler.log
stk30  -hwm 51  -mingain 60  -ignore  -mail  root
```

The following sections describe the parameters specified in Figure 11..

The `-hwm 51` *Parameter*

By specifying a high watermark, you can set the percentage of media usage below which recycling cannot occur. This percentage is the ratio of the used space in the library to its total capacity. As an example, a library that holds 10 20-gigabyte tapes, three of them 100 percent full and the remaining seven each 30 percent full, has the following media utilization percentage:

```
((3* 1.00 + 7 * 0.30) * 20G ) / ( 10 * 20G ) * 100%= 51%
```

Note that this calculation does not distinguish between current data and expired data. It only addresses the amount of media used.

In this example, if the high watermark is 51 percent or less, the recycler does not automatically select any of the automated library's VSNs for recycling.

Note: You can force a VSN to be recycled by using the following command to set the recycling flag:

```
# chmed +c lt.AAA123
```

When the `+c` flag is set, the archiver does not write any more archive images to the volume. The `+c` flag can be viewed through the `samu`(1M) utility. For more information, see the `chmed`(1M) and `samu`(1M) man pages.

The `-mingain=60` *Parameter*

The *minimum VSN gain percentage* sets a lower limit on the amount of space to be gained by recycling a cartridge. For example, if a cartridge in an automated library is 95 percent current data and 5 percent expired data, the gain obtained by recycling the cartridge is only 5 percent. It might not be worth moving the other 95 percent to retrieve this space. Setting the minimum-gain to 6 percent or more inhibits the recycler from automatically selecting this example VSN.

Another example is a cartridge with 90 percent expired data, 5 percent current data, and 5 percent free space. This would have a gain of 90 percent if recycled.

The `-ignore` *Parameter*

The `ignore` keyword keeps the recycler from recycling a particular library and should be used when you are configuring the recycler.

The `mail root` *Parameter*

The `mail` keyword specifies that the recycler send mail when recycling occurs on a given library. The mail message has the following subject line:

`Robot` ***robot-name*** `recycle`

Sample message bodies include the following:

`I will recycle VSN` ***vsn*****.**

`Cannot find any candidate VSN in this media changer.`

`Previously selected VSN` ***vsn*** `is not yet finished recycling.`

`Previously selected VSN` ***vsn*** `is now finished recycling. It will now be post-recycled.`

Step 2: Editing the archiver.cmd File (Optional)

Perform this step if you are recycling by archive set. If you are archiving to disk, recycling by archive set is the only means of recycling that is possible, so you must complete this step in order to recycle.

If you are recycling by library, you can proceed to the next step.

To recycle by archive set, edit the `/etc/opt/SUNWsamfs/archiver.cmd` file and add information for the archive sets you want to recycle. The recycling directives must appear between `params` and `endparams` directives. Table 35. shows the archive set recycling directives.

Table 35. Archive Set Recycling Directives

Directive	Function
`-recycle_dataquantity` *size*	Limits the amount of data that the recycler can schedule for rearchiving in its efforts to clear volumes of useful data.
`-recycle_hwm` *percent*	Sets the high watermark percentage.
`-recycle_ignore`	Prevents the archive set from being recycled.
`-recycle_mailaddr` *mail_address*	Sends recycler messages to *mail_address*.
`-recycle_mingain` *percent*	Limits recycling to those VSNs that would increase their free space by *percent* or more.
`-recycle_vsncount` *count*	Limits the number of volumes to be rearchived to *count*.

For more information on the preceding directives, see the `archiver.cmd`(4) man page.

As noted previously, include the `-recycle_ignore` directive in the global directives section to prevent the recycler from taking action before your configuration has been tested.

Figure 12. shows an `archiver.cmd` example for recycling disk archives.

Figure 12. Disk Archiving Specifications in the `archiver.cmd` File

```
fs = samfs1
    1 2m

arset0  testdir0
    1 2m
    2 4m

arset1  testdir1
```

Figure 12. Disk Archiving Specifications in the `archiver.cmd` File

```
    1 2m
    2 4m

params
arset0.1 -disk_archive disk01 -recycle_hwm 5 -recycle_mingain
2
arset1.1 -disk_archive disk02 -recycle_hwm 5 -recycle_mingain
2
endparams
```

Step 3: Running the Recycler

Run the `sam-recycler`(1M) command. The recycler reads the `recycler.cmd` file. Examine the standard output, log, ASM log, and `/var/adm/messages` for any error messages from the recycler. Figure 13. shows a sample recycler log file for recycling removable media cartridges.

Figure 13. Recycler Log File Example for Removable Media Cartridges

```
========== Recycler begins at Wed Dec 12 14:05:21 2001
===========
Initial 2 catalogs:

0  Family: m160                 Path: /var/opt/SUNWsamfs/
catalog/m160
   Vendor: ADIC                 Product: Scalar 100
   SLOT                 ty    capacity         space vsn
      0                 at        25.0G        25.0G CLN005
      1                 at        48.5G         6.1G 000003
      2                 at        48.5G        32.1G 000004
      3                 at        48.5G        35.1G 000005
      4                 at        48.5G        44.6G 000044
      5                 at        48.5G        45.1G 000002
      6                 at        48.5G        45.9G 000033
      7                 at        48.5G        48.5G 000001
   Total Capacity:  364.8G bytes, Total Space Available:
282.3G bytes
   Volume utilization 22%, high 95% VSN_min 50%
   Recycling is ignored on this robot.




1  Family: hy                   Path: /var/opt/SUNWsamfs/
catalog/historian
   Vendor: Sun SAM-FS           Product: Historian
   SLOT                 ty    capacity         space vsn
      (no VSNs in this media changer)
   Total Capacity:  0    bytes, Total Space Available: 0
bytes
```

Figure 13. Recycler Log File Example for Removable Media Cartridges

```
   Volume utilization 0%, high 95% VSN_min 50%
   Recycling is ignored on this robot.



8 VSNs:

                   ---Archives---   -----Percent-----   m160
 ----Status-----    Count    Bytes   Use Obsolete Free
Library:Type:VSN
no-data VSN            0      0        0    87      13
m160:at:000003
no-data VSN            0      0        0    33      67
m160:at:000004
no-data VSN            0      0        0    27      73
m160:at:000005
no-data VSN            0      0        0     8      92
m160:at:000044
no-data VSN            0      0        0     7      93
m160:at:000002
no-data VSN            0      0        0     5      95
m160:at:000033
empty VSN              0      0        0     0     100
m160:at:CLN005
empty VSN              0      0        0     0     100
m160:at:000001



Recycler finished.

========== Recycler ends at Wed Dec 12 14:05:32 2001
===========
```

Figure 14. shows a sample recycler log file for recycling disk archive files.

Figure 14. Recycler Log File Example for Disk Archive Files

```
---Archives---   -----Percent-----
 ----Status-----    Count    Bytes   Use Obsolete Free
Library:Type:VSN
new candidate          0      0        0    41      59
<none>:dk:disk01


677 files recycled from VSN disk01 (mars:/sam4/copy1)
0 directories recycled from VSN disk01 (mars:/sam4/copy1)
```

Step 4: Creating a crontab File for the Recycler (Optional)

If the system is performing as expected, you are ready to make a `crontab` entry for the superuser to run the recycler periodically. You might want to run the recycler no more than once every two hours, depending on your site's conditions.

The following example entry in root's `crontab` file ensures that the `cron` daemon runs the recycler every five minutes after the hour for every odd-numbered hour:

```
5 1,3,5,7,9,11,13,15,17,19,21,23   * * * /opt/SUNWsamfs/sbin/
                                               sam-recycler
```

Step 5: Removing -recycle_ignore and ignore Directives

Remove the `-recycle_ignore` directives from the `archiver.cmd` file, and remove the `ignore` directives from the `recycler.cmd` files. You are now recycling.

Step 6: Creating a recycler.sh File

Perform this step if you are recycling archive copies on removable media cartridges.

If you are archiving only to disk, do not perform this step.

The recycler executes the `recycler.sh` script when all the current images from a VSN have been rearchived to another VSN. The example found in `/opt/SUNWsamfs/examples/recycler.sh` and in Figure 15. shows how to relabel a recycled VSN and send mail to the superuser.

Figure 15. Example `recycler.sh` File

```
#!/bin/csh -f
#
#   /opt/SUNWsamfs/sbin/recycler.sh - post-process a VSN after
recycler has
#   drained it of all known active archive copies.
#
#   Arguments are:
#      $1 - generic media type "od" or "tp" - used to construct
the name
#           of the appropriate label command: odlabel or
tplabel
#
#      $2 - VSN being post-processed
#
#      $3 - Slot in the library where the VSN is located
```

Figure 15. Example `recycler.sh` **File (Continued)**

```
#!/bin/csh -f
#
#       $4 - equipment number of the library where the VSN is
located
#
#       $5 - actual media type ("mo", "lt", etc.) - used to
chmed
#            the media if required
#
#       $6 - family set name of the physical library, or the
string
#           "hy" for the historian library.   This can be used
to
#           handle recycling of off-site media, as shown below.
#
#       $7 - VSN partition, used for optical and D2 media
#
#
# $Id: recycler.sh,v 2.7 2000/04/10 14:51:45 ram Dev $
#

#   It is a good idea to log the calls to this script
#echo `date` $* >>  /var/opt/SUNWsamfs/recycler.sh.log

#   As an example, if uncommented, the following lines will
relabel the VSN,
#   if it exists in a physical library. If the VSN is in the
historian
#   catalog (e.g., it's been exported from a physical library
and moved
#   to off-site storage), then email is sent to "root"
informing that the
#   medium is ready to be returned to the site and reused.
#
#set stat=0
#if ( $6 != hy ) then
#    /opt/SUNWsamfs/sbin/chmed -R $5.$2
#    /opt/SUNWsamfs/sbin/chmed -W $5.$2
#    if ( $5 != "d2" ) then
#        if ( $1 != "od" ) then
#            /opt/SUNWsamfs/sbin/${1}label -w -vsn $2 -old $2
$4\:$3
#            if ( $status != 0 ) then
#                set stat = 1
#            endif
#        else
#            /opt/SUNWsamfs/sbin/${1}label -w -vsn $2 -old $2
$4\:$3\:$7
#            if ( $status != 0 ) then
#                set stat = 1
#            endif
```

Figure 15. Example `recycler.sh` **File (Continued)**

```
#!/bin/csh -f
#       endif
#    else
#        /opt/SUNWsamfs/sbin/${1}label -w -vsn $2 -old $2
$4\:$3\:$7
#       if ( $status != 0 ) then
#           set stat = 1
#       endif
#    endif
#else
#    mail root <</eof
#VSN $2 of type $5 is devoid of active archive
#images. It is currently in the historian catalog, which
indicates that
#it has been exported from the on-line libraries.
#
#You should import it to the appropriate library, and relabel
it using
#${1}label.
#
#This message will continue to be sent to you each time the
recycler
#runs, until you relabel the VSN, or you use the Sun SAM-FS
samu or
#robottool programs to export this medium from the historian
catalog to
#suppress this message.
#/eof
#endif
#echo `date` $* done >>  /var/opt/SUNWsamfs/recycler.sh.log
#if ( $stat != 0 ) then
#   exit 1
#else
#   exit 0
#endif
#
#
#   These lines would inform "root" that the VSN should be
removed from the
#   robotic library:
#
#mail root <</eof
#VSN $2 in library $4 is ready to be shelved off-site.
#/eof
#echo `date` $* done >>  /var/opt/SUNWsamfs/recycler.sh.log
#exit 0


#  The default action is to mail a message reminding you to set
up this
```

Figure 15. Example `recycler.sh` File (Continued)

```
#!/bin/csh -f
#  file. You should comment out these lines (through and
including the /eof
#  below) after you've set up this file.
#
mail root <</eof
The /opt/SUNWsamfs/sbin/recycler.sh script was called by the
Sun SAM-FS recycler
with the following arguments:

      Media type: $5($1)  VSN: $2  Slot: $3  Eq: $4
      Library: $6

/opt/SUNWsamfs/sbin/recycler.sh is a script which is called
when the recycler
determines that a VSN has been drained of all known active
archive
copies. You should determine your site requirements for
disposition of
recycled media - some sites wish to relabel and reuse the
media, some
sites wish to take the media out of the library for possible
later use
to access historical files. Consult the recycler(1m) man page
for more
information.
/eof
#echo `date` $* done >>  /var/opt/SUNWsamfs/recycler.sh.log
exit 0
```

The recycler called the `/opt/SUNWsamfs/sbin/recycler.sh` script with the following arguments:

```
Media type: $1  VSN: $2  Slot: $3  Eq: $4
```

The `/opt/SUNWsamfs/sbin/recycler.sh` script is called when the recycler determines that a VSN has been drained of all known active archive copies. You should determine your site requirements for dispensing with recycled cartridges. Some sites choose to relabel and reuse the cartridges; others choose to remove the cartridges from the automated library to use later for accessing historical files. For more information, see the `recycler`(1M) and `recycler.sh`(4) man pages.

Troubleshooting the Recycler

The most frequent problem encountered with the recycler is a message similar to the following. This message can be generated by the recycler when it is invoked:

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

This message can be caused by one of the following conditions:

- 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. This option displays the path names of the files associated with the 123 archive copies in the recycler log file, which are contained in messages similar to the following:

```
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.

Graphical User Interface (GUI) Tools

8

This chapter describes the GUI tools used to manage the devices in a ASM or ASM-QFS environment. Two GUIs are used to manage robots, devices, and media mount requests:

- `libmgr`(1M)—This provides a single interface to all automated libraries and devices and can be customized for operations at your site.
- `samtool`(1M)—This consists of three interfaces: `robottool`, `devicetool`, and `previewtool`.

To use the tools, you need to be familiar with a window system and be familiar with mouse buttons.

Note: In addition to these GUIs, you can manage devices in a ASM or ASM-QFS environment using the full-screen operator tool, samu(1M). For instructions on using the `samu`(1M) utility, see "Using the samu(1M) Operator Utility" on page 223.

This chapter contains the following topics:

Operator Privilege Levels

The GUI tools described in this chapter are designed for use by superusers only. The `libmgr`(1M) can be executed only by superusers. The `samtool`(1M), `robottool`(1M), `devicetool`(1M), and `previewtool`(1M) interfaces, however, can be executed by superusers and by individuals included in an operator group.

As a site administrator, you can define operational authority that does not grant superuser privileges. This special authority, however, grants the ability to perform operator-type functions, such as clearing tape load requests and changing device states. You can set up an operator group and define permissible operator tasks in the `/etc/opt/SUNWsamfs/defaults.conf` file. Users with root authority have full access to functions within `samtool`. Users who are part of the operator group have limited access and can perform only certain functions. This becomes apparent when attempting to use the functions within `robottool`, `devicetool`, and `previewtool`.

A single operator group is defined in the `defaults.conf` file using the `operator` keyword. Privileged tasks for the operator group are defined using the `oper_privileges` keyword. Labeling of media, performing audits, moving cartridges in an automated library, and changing device states are all examples of operator tasks that can be defined.

For a complete listing of operator-privileged tasks, see the `defaults.conf`(4) man page.

■ Using libmgr(1M)

The library manager (invoked with the `libmgr`(1M) command) is a GUI tool for managing automated libraries. You can use `libmgr`(1M) to check the status of automated libraries and cartridges, to import and export cartridges, and to respond to cartridge load requests.

To Start the Library Manager

To start the library manager, enter the following command at the operating system prompt:

```
# libmgr&
```

To Reset Library Manager Displays, Images, and Titles

The displays, images, and titles in `libmgr` are highly configurable. Upon startup, `libmgr` reads the `/etc/opt/SUNWsamfs/SamGUI.rsc` resources file. Without any changes, `libmgr` displays device titles and images based on the device's product ID, vendor ID, and Equipment Number as defined in the `mcf` file.

You can use the `SamGUI.rsc` file to set the following:

- Device and media titles and images
- Catalog settings
- Mount request settings
- Screen settings including height, width, and font sizes

For a complete listing of resource settings, see the `SamGUI.rsc`(4) man page.

To reconfigure the `libmgr` displays, you must edit the `SamGUI.rsc` file, exit `libmgr`, and restart `libmgr`.

Library Manager Display

The library manager display is divided into three horizontal panels, as follows:

- The libraries panel is on top. This panel does not appear if you do not have any robots configured.
- The catalog panel is in the middle.
- The file systems and mount request panel is on the bottom.

The display consists of objects that can be manipulated by a mouse. Table 36. shows how most objects respond to the mouse.

Table 36. Mouse Actions

Mouse Operation	Behavior
Left click	Selects an object.
Right click	Displays a pull-down menu of actions.
Double click	Displays detailed information regarding the object.

To Perform Robot Operations

In this panel, place the pointer on the desired robot image. Table 37. shows the actions you can perform.

Table 37. Robot Operations

Desired Action	Mouse Button	Menu Pick
Turn automated library to on, off, or down.	Right-click	Choose On, Off, Unavailable, or Down.
Import media.	Right-click	Choose Import.
Unload VSNs from the robot catalog.	Right-click	Choose Unload. The robot’s catalog is emptied and the robot is set to off. Set the robot to on to reset.
Fully audit the robot.	Right-click	Choose Audit.

To Perform Media Operations

In the catalog panel, select the desired robot. Select the media with a left click, then choose from the actions in Table 38..

Table 38. Media Operations

Desired Action	Mouse Button	Menu Pick
Label or relabel media.	Right-click	Choose Label. Enter VSN, blocksize, and optionally relabel or erase.
Audit VSN.	Right-click	Choose Audit.
Mount VSN.	Right-click	Choose Mount.
Move VSN.	Right-click	Choose Move. Enter destination slot number.
Export VSN.	Right-click	Choose Export.

To Perform Media Drive Operations

Place the pointer on the desired media drive image, and then choose from the actions shown in Table 39..

Table 39. Media Drive Operations

Desired Action	Mouse Button	Menu Pick
Label or relabel media.	Right-click	Choose Label. Enter VSN, blocksize, and optionally relabel or erase.
Turn drive to on, off, unavailable, or down.	Right-click	Choose On, Off, Unavailable, or Off.

To View File System States and Attributes

To view the file system states and attributes and make changes, double-click on the desired file system. A detailed information window for this file system is displayed.

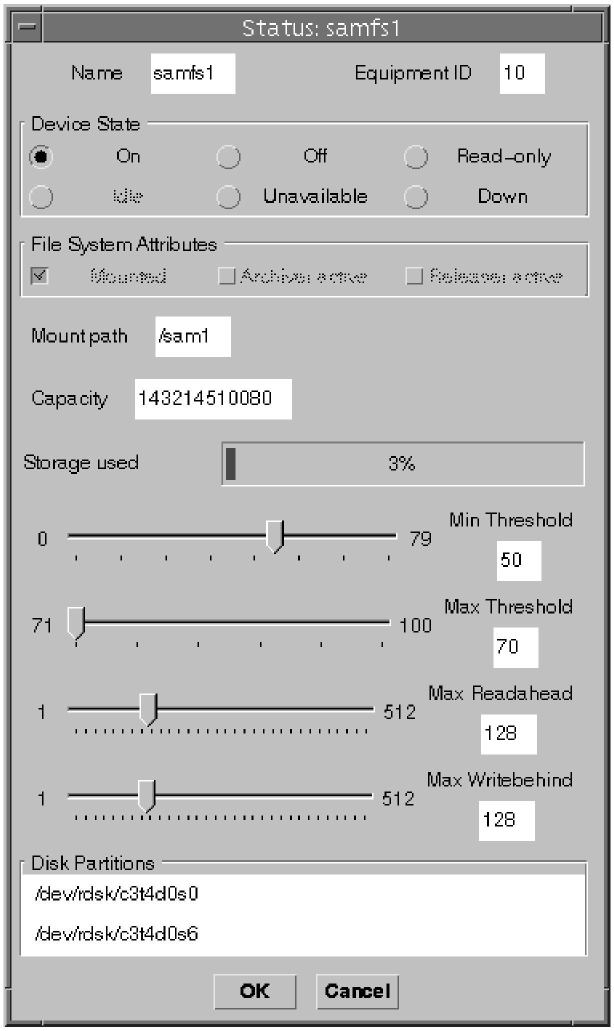

Figure 16. `libmgr`**(1M) — File System States and Attributes**

The file system device states and device attributes are described on the `libmgr`(1M) man page under the Icon Attributes heading. You can change the attributes or states by selecting the button and clicking OK.

Note: Any changes to mount options made in this interface persist only until the file system is unmounted.

To View Media Drive States and Attributes

To view the media states and attributes and make changes, double-click the desired media drive image. The device state and attributes window for this device is displayed.

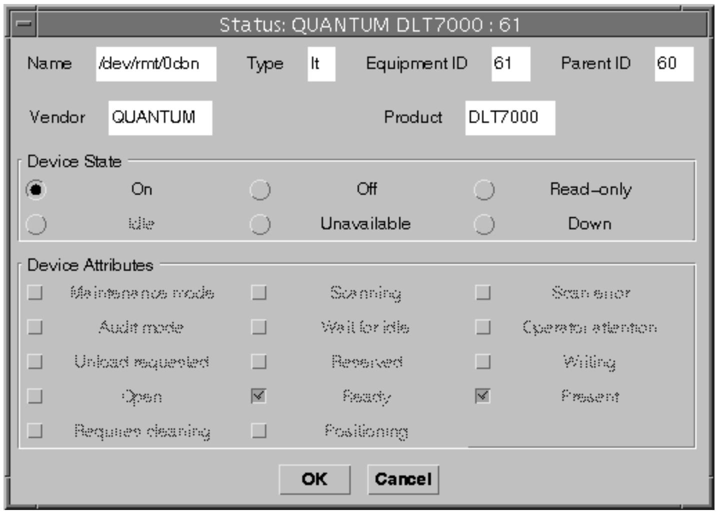

Figure 17. `libmgr`**(1M) — Media Drive States and Attributes Screen**

The media drive device states and device attributes are described in the `libmgr`(1M) man page under the Icon Attributes heading. Any changes to the attributes or states are enabled by selecting the appropriate button and clicking OK.

To View the VSN Catalog Display

To view the catalog settings for a VSN, double click on the desired VSN in the catalog panel. A detailed list of attributes for the selected VSN is displayed. This information is derived from the robot catalog as defined in the `mcf` file.

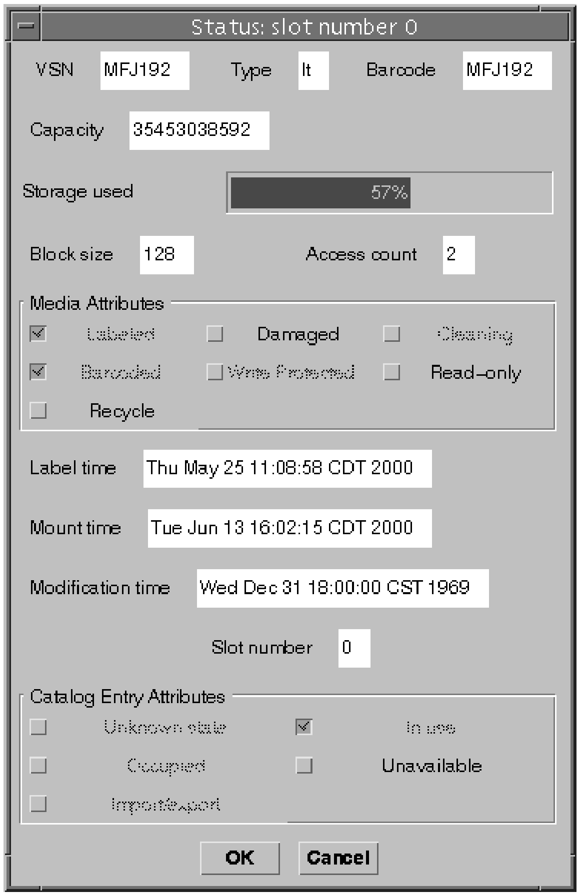

Figure 18. `libmgr`**(1M) — VSN Catalog Display**

■ Using samtool(1M)

The `samtool`(1M) is the initial launch window for `robottool`(1M), `devicetool`(1M), and `previewtool`(1M). The following sections describe these tools.

To Start and Quit samtool(1M)

To start `samtool`, enter the following command at the operating system prompt:

```
# samtool&
```

The system displays the `samtool` group. This display includes icons for `robottool`(1M), `devicetool`(1M), and `previewtool`(1M).

To exit `samtool`, right-click the top bar of the window and select QUIT or Close.

To Select a Tool

The `samtool`(1M) display includes an icon for each of its tools. To invoke a tool, left-click on the icon of the tool you want to use. The tools are as follows:

- `robottool`(1M) enables you to view and manage information on robots configured within the ASM or ASM-QFS environment.
- `devicetool`(1M) enables you to view and manage information on devices configured within the ASM or ASM-QFS environment.
- `previewtool`(1M) enables you to view and manage pending mount requests within the ASM or ASM-QFS environment.

Note: You can also start a tool by entering the tool's name on the command line. For example, to start `robottool`(1M), enter `robottool` on the command line. To start a tool in the background, type an ampersand (`&`) after the tool name. For example, to start `robottool`(1M) in the background, enter `robottool&`.

To Update the Displays

By default, all `samtool`(1M) displays are automatically refreshed every five seconds. You can change the refresh rate or disable automatic refresh. You can also force the display to update itself when necessary. The update button, the refresh checkbox, and the refresh field control the updates.

To Change the Refresh Rate

1. Make sure that automatic refresh is enabled.

   That is, make sure that the refresh checkbox contains a check mark to indicate that it is enabled.

2. Type a new refresh rate in the refresh field or use the increase/decrease setting buttons.

To Update a Tool Display

- To immediately update a tool display, click the `Update` button located in the upper right of the window.

To Control Automatic Refresh

- To enable or disable automatic refresh, click the refresh.

  Automatic refresh is enabled when the refresh checkbox contains a check mark.

To Manage Screen Resources

You can change the font used for panel lists in `samtool`(1M) displays by using the `fontfamily` resource setting. The following example from a `.Xdefaults` resource file defines a font family to be used with `robottool`:

```
robottool.fontfamily: fixed
```

To Display Online Help

1. Left click on the Help button to display general help on `samtool`(1M) and its operation.
2. Right click on the Help button to display a menu containing an item for each tool.

   Select the menu item that corresponds to the tool for which you want help.

Using robottool(1M)

The `robottool` utility displays configured robots, the VSN catalog associated with a selected robot, and the devices associated with a selected robot. By default, when `robottool` is started, the first ASM or ASM-QFS robot is selected. Selecting a robot in the robot display causes the system to display the VSN catalog and devices for the selected robot.

Figure 19. shows a sample `robottool` display.

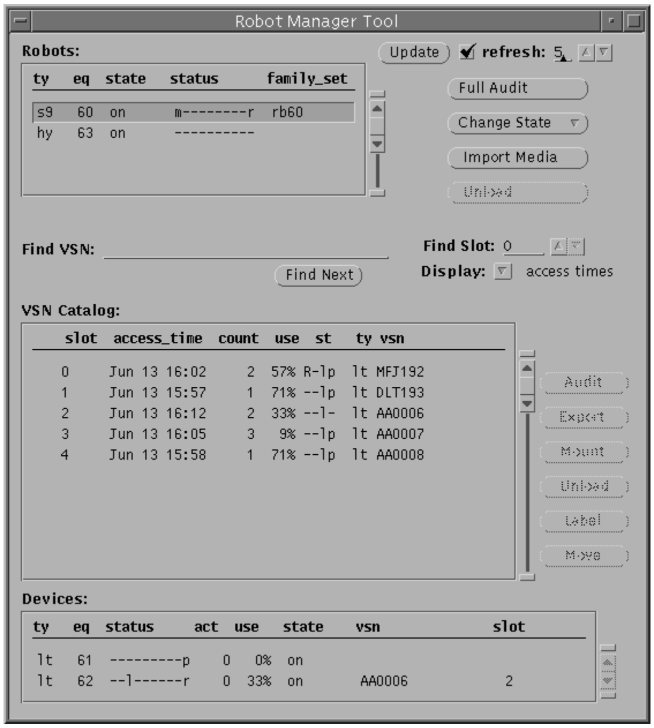

Figure 19. `robottool`**(1M) — Initial Screen**

The `robottool` display contains three areas:

- Robots
- VSN Catalog
- Devices

The following sections describe the content of these display areas.

To Start robottool(1M)

Enter the following command at the operating system prompt:

```
# robottool&
```

Robots

The Robots area lists all robots configured within a ASM or ASM-QFS environment. Table 40. shows the information that is displayed for each robot.

Table 40. Robot Display

Information	Description
`ty`	Equipment Type.
`eq`	Equipment Ordinal.
`state`	State of equipment. See “To Change the State of a Robot” on page 208.
`status`	Status of the robot. See “Viewing Status Information” on page 206.
`family_set`	Name of the Family Set to which the robot belongs.

You can use the robot buttons to perform a full audit of all media, change the state of the robot, import media, and unload media.

VSN Catalog

The VSN Catalog area lists the VSNs for the selected robot. Table 41. shows the information that is available for each VSN.

Table 41. VSN Catalog Display

Information	Description
`slot`	Slot number of the media.
`access_time`	Time at which the media was last accessed.
`barcode`	Barcode for the media.
`count`	Number of times the media has been accessed.
`use`	Percentage of used space for the media.
`st`	Status of the VSN. For more information, see “Viewing Status Information” on page 206.
`ty`	Media type.
`vsn`	Volume serial name.

The VSN display includes either access time or barcode information for all VSNs. You can use the VSN action buttons to audit, export, mount, unload, label, and move volumes.

Devices

The devices area displays the information on devices for the selected robot. Table 42. shows the information that is displayed.

Table 42. Devices Display

Information	Description
`ty`	Equipment Type.
`eq`	Equipment Ordinal.
`status`	Status of the device. See “Viewing Status Information” on page 206.
`act`	Activity counter.
`use`	Percentage of used space for the volume mounted in the device.
`state`	State of the device.
`vsn`	Volume Serial Name of the medium.
`slot`	Slot number of the medium.

To control devices, see “Using devicetool(1M)” on page 214.

Viewing Status Information

Table 43. describes the status bits.

Table 43. Status Bits

Status Bit	Meaning for Device	Meaning for File System
`s---------`	Media is being scanned.	
`m---------`	The file system is currently mounted.	
`M---------`	Maintenance mode.	
`-E--------`	Device received an unrecoverable error in scanning.	
`-a--------`	Device is in audit mode.	The file system is being archived.
`--l-------`	Media has a label.	

Table 43. Status Bits

Status Bit	Meaning for Device	Meaning for File System
`--N-------`	Media is foreign to the ASM or ASM-QFS environment.	
`---I------`	Waiting for device to idle.	
`---A------`	Needs operator attention.	
`----C-----`	Cleaning cartridge.	
`----U-----`	Unload has been requested.	
`-----R----`	Device is reserved.	
`------w---`	A process is writing on the media.	
`-------o--`	Device is open.	
`--------P-`	Device is positioning (tape only).	
`--------F-`	All storage slots occupied (robot status only).	
`---------W`	Device is ready and media is write protected.	
`---------R`	Device is ready and the media is read only.	
`---------r`	Device is spun up and ready.	The file system’s disk space is being released.
`---------p`	Device is present.	

Managing Robots

This section describes the actions you can perform on a selected robot. Table 44. shows the possible actions.

Table 44. Managing Robots

Action	Description
Full audit	Perform a full audit for all volumes in the selected robot.
Change state	Change the state of the robot.
Import media	Import media into the selected robot.
Unload	Unload all media from the selected robot.

To Perform a Full Audit

1. Select the robot in the list of available robots.
2. Left-click the Full Audit button. You are prompted to confirm the operation.

The system performs a full audit of every volume in the robot.

To Change the State of a Robot

1. Select the robot in the list of available robots.
2. Do one of the following:

- Left-click on the Change State button to change the state to ON.
- Right-click on the Change State button to display a list of states. Table 45. shows some of the possible states.

Table 45. Changing a Robot State

Current State	Possible Next State
ON	IDLE, OFF
IDLE	Automatically goes to OFF when IDLE
OFF	DOWN, ON
DOWN	OFF

Importing and Exporting Media

To Import Media Into a Robot

1. Select the robot in the list of available robots.
2. Click the Import Media button.
3. Place the cartridge in the robot's mailbox.

The system instructs the robot to accept the cartridge placed in the robot's mailbox. When you have selected Import Media you can continue to place cartridges in the mailbox. If 30 seconds pass and no cartridges have been inserted, the import operation is terminated.

To Export Cartridges Out of a Robot

1. Select the robot in the list of available robots.
2. Select the slot from which you want to export.
3. Click the Export Media button.

The system instructs the robot to place the selected cartridge into the robot's mailbox.

Note: You can import and export cartridges only when the robot device has a mailbox.

Loading and Unloading Magazines

To Load a Magazine

1. Select the robot in the list of available robots.

   The selected robot must not have a magazine currently loaded.
2. Click the Load button.

The system instructs the robot to load the magazine.

To Unload a Magazine

1. Select the robot in the list of available robots.

   The selected robot must have a magazine currently loaded.
2. Click the Unload button.

The system instructs the robot to unload the magazine.

Note: You can load and unload a magazine only when the selected robot supports loading and unloading magazines.

Working With Volumes

When a robot is selected, all the volumes for that robot are displayed in the VSN catalog located in the middle of the screen. This section provides instructions for working with volumes.

To Display Barcodes Rather Than Access Times

The catalog display includes information for each slot in the selected robot. You can include either access times or barcodes in the catalog display. By default, access times are displayed.

- To display barcodes rather than access times, right-click on the Display button and select barcodes.

The system displays barcodes rather than access times.

To Find a VSN

To search for and select a volume by VSN, perform the following step.

- Type a VSN name or beginning pattern and press the Return key to match in the Find VSN field.

  If the system finds the VSN pattern you have specified, the first VSN of the specified pattern is selected. To find the next occurrence of the specified pattern, click the Find Next button. If no VSN is found, an error message is generated.

  When searching for VSNs using a pattern match, a VSN is considered a match if the pattern of length *n* characters that is entered exactly matches the first *n* characters of the VSN.

To Find a VSN Within a Specific Slot Number

- Type a slot number in the Find Slot field.

You can also use the up and down buttons to increase or decrease the slot number. If no VSN is found, an error message is generated.

To go to the next slot number, click the Find Next button.

To Select a VSN

- Left-click on the VSN you want to select.

To Audit a VSN

To perform an audit on a selected VSN, perform the following steps.

1. Select the VSN for which you want to perform an audit.
2. Left-click on the Audit button.

The system reads the VSN and updates the catalog entry for the slot.

Note: To perform an audit for every VSN in a robot, select the robot in the Robot Display and click the Full Audit button.

To Export a Volume

1. Select the VSN you want to export. The VSN you select must currently be in the robot.
2. Left-click on the Export button.

The robot removes the VSN and places it in the robot mailbox.

To Load a Volume

1. Select the VSN you want to load.
2. Left-click on the Mount button.

The robot mounts the selected VSN into one of the robot's devices.

To Unload a Volume

1. Select the VSN you want to unload.
2. Left-click on the Unload button.

The robot unloads the selected VSN from the robot's device and puts it back in the slot.

To Label a Volume

Within the ASM and ASM-QFS environments, labeled volumes distinguish one cartridge from another. A software label provides the ASM and ASM-QFS software with important information, including the name of the VSN and the location at which data begins to be written on the cartridge.

Caution: Labeling a volume causes the loss of previously written data on that volume. Be sure that this is your intention before you proceed.

Perform the following steps to label a volume.

1. Select the VSN you want to label.
2. Left click on the Label button.

   Figure 20. shows the dialog box that is displayed.

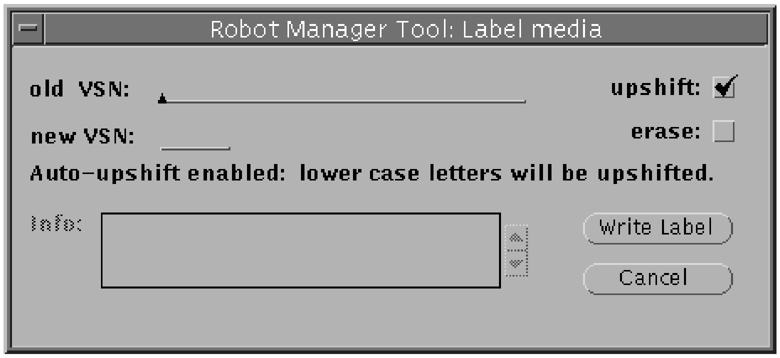

Figure 20. `robottool`**(1M) — Media Labeling Screen**

3. Do one of the following.

For an old VSN, if you are relabeling a volume, type the old VSN. The old VSN must exactly match the volume's current VSN. If you want the tool to automatically shift lowercase letters to uppercase, click the Upshift box. If you are relabeling a tape and Upshift is selected, the old VSN can differ in case from the tape's current VSN.

For a new VSN, type a new VSN. For optical media, the VSN can be up to 31 characters. For all other media, the VSN can be up to 6 characters. For optical media, you can type up to 128 characters in the Info window for inclusion in the label.

4. If you want to erase the media during the labeling operation, click the erase box.

   Erasing media can require a significant amount of time. Note that data is always lost during a label operation. Erasing overwrites each sector on the volume.

5. Click the Write Label button.
6. If an error is detected, an error checkbox and message appear in the Label media window above the Info box.

   To acknowledge the error, click in the checkbox, and the error message is removed.

Possible errors include an invalid VSN or an old VSN that does not match the VSN of the volume in the selected slot.

To Move Media

Perform the following steps to move a volume to another slot.

1. Select the volume you want to move.
2. Left-click on the Move button.

   Figure 21. shows the dialog box that is displayed.

Figure 21. `robottool`**(1M) — Media Moving Screen**

3. Enter the following in this box.

For the Source slot, enter a new slot number by either double-clicking the slot number and typing a new number or by backspacing over the number to erase it and typing a new number. The Source slot you specify must contain a volume. By default, the Source slot field contains the slot number of the volume you have selected

For the Destination slot, type a new Destination slot number. The slot number you specify must be available

4. Click the Move button.
5. If an error is detected, an error checkbox and message appear in the Move media window.

   To acknowledge the error, click in the checkbox, and the error message is removed.

Possible errors include not specifying a source or destination slot, or specifying a slot that is not valid. A valid slot is an integer greater than or equal to zero and less than the number of entries in the robot's catalog.

Viewing VSN Status Information

The `st` column displays the status of the catalog entry. Table 46. shows the possible status bits.

Table 46. VSN Status Bits

Status Bit	Meaning
`A---`	Volume needs audit.
`R---`	Volume is marked for recycling.
`W---`	Volume is write protected.
`-E--`	Bad media.
`-X--`	This is an export slot.
`-r--`	Volume is marked read-only.
`--u-`	Slot is unavailable.
`--l-`	Volume is labeled.
`--N-`	Volume is foreign to the ASM or ASM-QFS environment.
`---c`	Cleaning.
`---p`	Slot is occupied.

Viewing Device Information

The lower third of `robottool` displays devices associated with the selected robot. This display is for information only and does not allow actions to be performed on the devices. To manage individual nonrobotic devices, use `devicetool`. The information displayed is the same as that displayed in the `devicetool` media-specific displays.

■ Using devicetool(1M)

The `devicetool`(1M) program is a GUI tool for viewing information about and managing devices associated with ASM and ASM-QFS.

To Start devicetool(1M)

To start `devicetool`, enter the following command at the operating system prompt:

```
# devicetool&
```

Figure 22. shows an initial `devicetool`(1M) display.

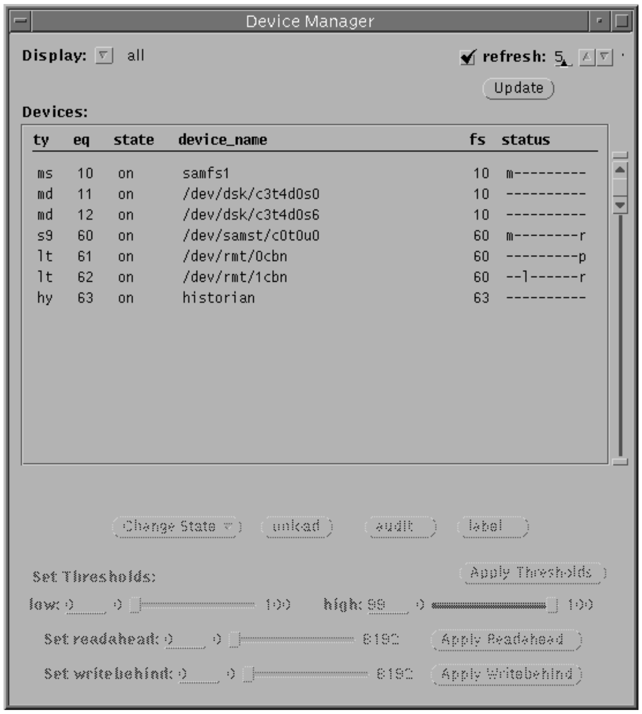

Figure 22. `devicetool`**(1M) — Initial Screen**

To Change the Display Format

The `devicetool`(1M) displays individual devices configured within the ASM or ASM-QFS environment. The devices are displayed in a scrollable list in the center of the screen. By default, all devices are displayed.

Perform the following steps to change the display format.

1. Right-click MENU on the Display button.

   A pull-down menu is displayed.

2. Select a display option.

3. If you have selected Removable Media or Manual Only, a Media pull-down menu button is displayed.

   To further restrict the display, right-click the Media button. A pull-down menu is displayed.

4. Select a media type for the display.

To View the Display Fields

Some display fields are common to all displays. Other fields are displayed only for certain display formats. Table 47. lists all the possible fields displayed in alphabetical order.

Table 47. Display Fields

Field	Description
`act`	Activity count. The number of times the volume is opened.
`device_name`	Name assigned to the drive.
`eq`	Equipment Ordinal of the device.
`family_set`	Name of the Family Set to which the device belongs.
`free`	Number of 1024-byte blocks of disk space available.
`fs`	Name of the file system to which the device belongs.
`low/high`	Low and high disk usage thresholds percentage.
`ord`	Ordinal number of the disk device within the storage family set.
`ra`	Maximum readahead on this file system in units of 1-kilobyte blocks. This is truncated to a multiple of 8 kilobytes.
`state`	Current operating state of the device. Valid device states are as follows: • `ready`—The device is on and the disk or tape loaded in the transport is available for access. • `notrdy`—The device is on but no disk or tape is present in the transport. • `idle`—The device is not available for new requests. Operations in progress continue until completion. • `off`—The device is n ot available for access. • `down`—The device is available only for maintenance access.
`status`	Device status.
`ty`	Device type.
`used`	Percentage of space used.
`vsn`	Volume serial number assigned to the volume, or `nolabel` if the volume is not labeled. This field is blank if no cartridge is present in the transport or if the device is off.
`wb`	Maximum writebehind on this file system in units of 1-kilobyte blocks.

Controlling Devices

The following sections provide instructions for controlling devices. They address the following topics:

- Changing the device state
- Unloading a device
- Auditing a volume in a device
- Labeling a volume in a device
- Setting thresholds
- Setting readahead and writebehind

To Select a Device

- To select a device from the display, click SELECT on the line representing the device.

When a device is selected, the buttons for actions appropriate for that device type are activated below the display. Depending on the device, you can change its state, unload, audit, and label.

To Change the Device State

You can use the Change State button to change the device state. Clicking SELECT on this button results in the default state, on, being selected. Clicking MENU on this button displays the Change State menu on which you can select a device state. Possible states are on, idle, off, and down. To change a device's state, perform the following steps.

1. Select the device in the list of available devices.
2. Do one of the following:
   - Left-click on the Change State button to change the state to ON.
   - Right-click on the Change State button to display a list of states. Table 48. shows the possible device states.

Table 48. Possible Device States

Current State	Possible Next State
ON	IDLE, OFF
IDLE	Automatically goes to OFF when IDLE
OFF	DOWN, ON
DOWN	OFF

To Unload a Device

1. Select the device you want to unload.
2. Left-click on the Unload button.

The robot unloads the selected device.

To Audit a Device

1. Select the device for which you want to perform an audit.
2. Left-click on the Audit button.

The system reads the volume in the device and updates the library catalog entry.

Note: To perform an audit for every VSN in a robot, select the robot in the Robot Display and click the Full Audit button.

To Label a Volume in a Device

1. Select the device for which you want to label media.
2. Left-click on the Label button.

   Figure 23. shows the dialog box that is displayed.

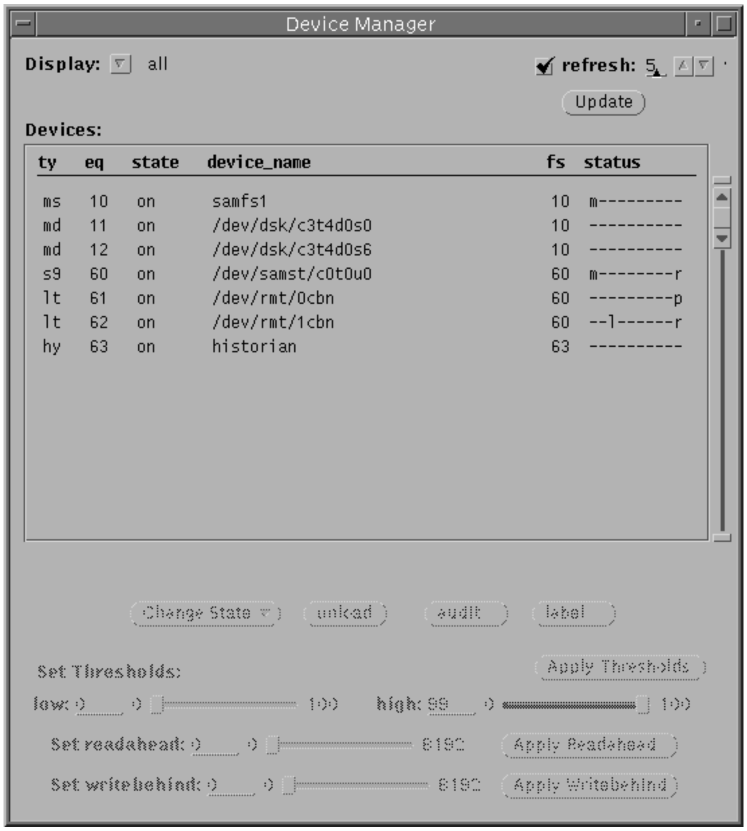

Figure 23. `devicetool`**(1M) — Labeling Media Screen**

3. Do one of the following:

- For an old VSN, if you are relabeling a volume, type the old VSN. The old VSN must exactly match the volume's current VSN. If you want to automatically shift lowercase letters to uppercase, click the upshift box. If you are relabeling a tape and upshift is selected, the old VSN can differ in case from the tape's current VSN.
- For a new VSN, type a new VSN. For optical media, the VSN can be up to 31 characters, and you can type up to 128 characters in the Info window for inclusion in the label. For all other media, the VSN can be up to 6 characters.

4. If you want to erase the volume during the labeling operation, click the erase box.

   Erasing media can require a significant amount of time.

5. Click the Write Label button.

If an error is detected, an error checkbox and message appear in the Label media window above the info box. To acknowledge the error, click the checkbox, and the error message is removed.

Possible errors include an invalid VSN or an old VSN that does not match the VSN of the media in the selected slot.

Caution: Labeling a volume causes the loss of all data on that volume.

To Set Thresholds

For disk sets, you can set low and high thresholds for the disk set by performing the following steps.

1. Select the disk set for which you want to set thresholds.
2. Type a number that specifies the percentage of use for either the low threshold or the high threshold, whichever one you are setting.

   Alternatively, you can use the slider bar to increase or decrease the number.
3. Click SELECT on the Apply Thresholds button.

   The new thresholds remain in effect until changed or until the file system is remounted.

To Set Readahead and Writebehind

You can set the maximum number of contiguous 1-kilobyte blocks for readahead and writebehind on a disk set. Perform the following steps to set readahead or writebehind.

1. Select the disk set for which you want to set readahead or writebehind.
2. Type a number that specifies the number of contiguous 1-kilobyte blocks for either reading ahead or writing behind, whichever one you are setting.

   Alternatively, you can use the appropriate slider bar to increase or decrease the number.
3. Click SELECT on the Apply Readahead or Apply Writebehind button.

   The new setting for readahead or writebehind remains in effect until it is changed or until the file system is remounted.

■ Using previewtool(1M)

The `previewtool`(1M) program allows you to view and manage pending mount requests. Initially, the display shows all pending mount requests in the mount request window. The information is displayed as a scrolling list. The window can also be resized to show from one to 18 mount requests by grabbing the window corner and stretching or contracting.

Figure 24. shows an initial `previewtool`(1M) display.

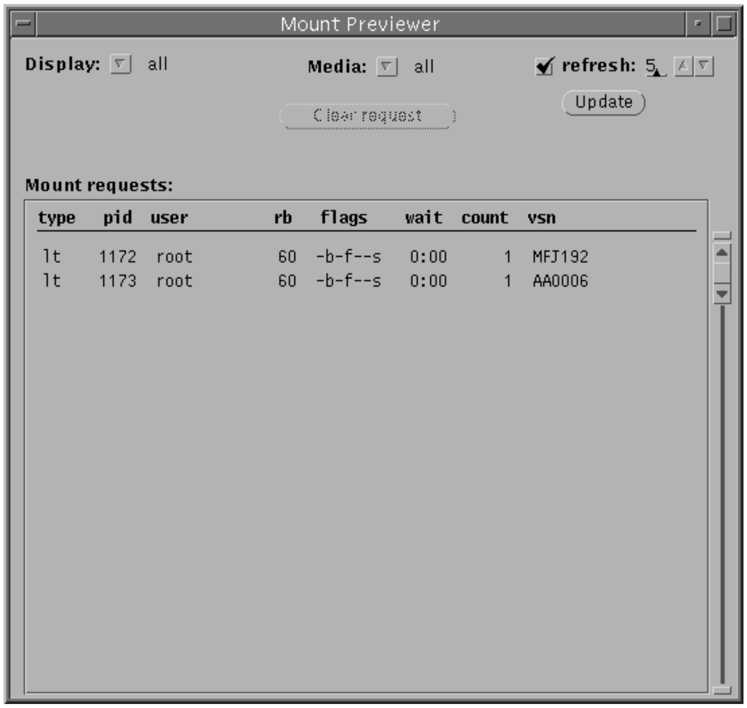

Figure 24. `previewtool`**(1M) — Initial Screen**

To Change the Display Format

1. Right-click on the Display button.
2. Select one of the options displayed.
3. Select a display type, as follows.

- If you select a specific robot, a menu of available robots is displayed. Select the robot you want to display.
- If you make a selection other than a specific robot, you can further restrict the display by media type. Right-click on the Media button. Select a media type to which you want to restrict the display.

The previewtool(1M) Display Fields

Table 49. shows the `previewtool`(1M) display fields.

Table 49. The `previewtool`(1M) Fields

Field	Description
`slot`	Slot number of the volume.
`type`	Device-type code assigned to the volume.
`pid`	UNIX process identifier. A process identifier of `1` indicates NFS access.
`user`	Name assigned to the user requesting the mount.
`rb`	Equipment Ordinal of the robot in which the requested VSN resides.
`flags`	See Table 50. for the description of the `flags` field.
`wait`	Elapsed time since the mount request was received. If the time is greater than one day, the time is displayed in days; otherwise, the time is displayed as *hh*:*mm*.
`count`	If the request is a stage mount, displays the number of requests for this VSN.
`vsn`	Volume serial number assigned to the media.

Table 50. explains the `flags` field.

Table 50. Flag Definitions

Flag	Meaning
`W------`	Write access requested
`-b-----`	Entry is busy
`--C----`	Clear VSN requested
`---f---`	File system requested
`----B--`	Use block I/O for data transfers
`-----S-`	Flip side already mounted
`------s`	Stage request flag

To Clear a Mount Request

1. Select the VSN for which you want to clear the request.
2. Left-click on the Clear request button.

Using the samu(1M) Operator Utility

9

This chapter provides instructions for controlling the devices configured within your ASM/QFS-Standalone, ASM, and ASM-QFS environment through the samu(1M) operator utility. Not all samu(1M) displays are supported in an ASM/QFS-Standalone environment, but the three types of environments are described in this chapter for the sake of completeness.

The following topics are presented:

Overview

The samu(1M) operator utility requires a display terminal that displays a minimum of 24 lines by 80 characters wide. The utility includes the following features:

- Displays that enable you to monitor ASM/QFS-Standalone, ASM, and ASM-QFS devices and file system activity
- Commands that enable you to select displays, set display options, control access to and the activity of devices, and take snapshots of display windows

The display windows shown in this chapter are representative examples. The exact format and amount of information displayed on your terminal can be different depending on your terminal model and the devices configured in your ASM/QFS-Standalone, ASM, or ASM-QFS environment.

The operations that can be performed from within samu(1M) can also be performed by using the samcmd(1M) command. For more information on samcmd(1M), see the samcmd(1M) man page.

The following sections describe how to start and stop samu(1M), interact with the utility, access the help windows, and view operator displays.

To Invoke samu(1M)

1. To start `samu`(1M), enter the `samu`(1M) command from the UNIX command line as follows:

```
# samu
```

The system starts `samu`(1M) and shows the help display.

2. Press CTRL-f to move to the next help screen, which shows the keys that control the displays.

The `samu`(1M) command accepts options on its command line. These options include those for selection of its initial display. For more information on the `samu`(1M) command line options, see the `samu`(1M) man page.

Note: `samu`(1M), like the `vi`(1) editor, is based on the `curses`(3X) library routine. You must have your terminal type defined correctly before invoking `samu`(1M).

To Stop samu(1M)

- To exit `samu`(1M), enter one of the following:
- Press the `q` key
- Enter `:q`

The `samu`(1M) operator utility exits and returns you to the command shell.

Interacting With samu(1M)

Interacting with `samu`(1M) is similar to interacting with the UNIX `vi`(1) editor with respect to paging forward or backward, entering commands, refreshing the display, and quitting the utility.

While viewing an operator display, you can use the keys described in Table 51. to control the display. The exact function of these keys depends on the display being viewed at the time. For information on display-specific key operations, see the `samu`(1M) man page.

Table 51. `samu`**(1M) Display Control Key Sequences**

Key	Function	Display
CTRL-b	Previous file system	`:a,a`
	Page backward	`c,h,o,p,s,t,u,v,w`
CTRL-d	Half-page forward	`c,p,s,u,w`
	Next robot catalog	`v`

Table 51. samu(1M) Display Control Key Sequences (Continued)

Key	Function	Display
	Page forward (top portion)	h
	Page forward (bottom portion)	a
CTRL-f	Next file system	:a,a
	Page forward	c,h,o,p,s,t,u,v,w
CTRL-k	Select (manual, robotic, both, priority)	p
	Advance sort key	v
	Toggle path display	n,u,w
CTRL-u	Half-page backward	c,p,s,u,w
	Previous robot catalog	v
	Page backward (top portion)	h
	Page backward (bottom portion)	a
CTRL-i	Detailed, 2-line display format	v
1-7	Select sort key, as follows: • 1 sorts by slot. • 2 sorts by count. • 3 sorts by usage. • 4 sorts by VSN. • 5 sorts by access time. • 6 sorts by barcode. • 7 sorts by label time.	v
/	Search for VSN	v
%	Search for barcode	v

Command and display error messages are displayed on the last line of the display window. If a command error occurs, automatic display refreshing halts until the next operator action.

Entering a Device

Each device included in the ASM/QFS-Standalone, ASM, or ASM-QFS environment is assigned an Equipment Ordinal (for example, 10) in the mcf file. Many samu(1M) commands reference a specific device.

Example 1. The syntax for the `:off` command is as follows:

`:off` *eq*

For the *eq*, enter the Equipment Ordinal for the device you are trying to address.

Example 2. At certain times, `samu`(1M) prompts for a device to be entered. When you access the Robot Catalog Display (described later in this chapter), you are prompted to enter a robot Equipment Ordinal:

```
Enter robot:
```

At the prompt, enter the Equipment Ordinal, or enter a carriage return to select the previous device used.

Getting Online Help

When you start `samu`(1M), the system automatically displays the first help screen. This help screen differs depending on whether you have a ASM/QFS-Standalone, ASM, or ASM-QFS file system. There are five pages of help screens, but this manual shows only the first. Subsequent help screens show `samu`(1M) commands.

For the ASM and ASM-QFS file systems, Figure 25. shows the initial help screen.

Figure 25. ASM and ASM-QFS `samu`(1M) Initial Help Screen

```
Help information            page 1/5   samu 4.0-x Thu Oct 11
13:22:30

Displays:
    a Archiver status                   v Robot catalog
    c Device configuration              w Pending stage queue
    d Daemon trace controls             C Memory
    f File systems                      F Optical disk label
    h Help information                  I Inode
    l License information               J Preview shared memory
    m Mass storage status               L Shared memory tables
    n Staging status                    M Shared memory
    o Optical disk status              N File System Parameters
    p Removable media load requests     R SAM-Remote
    r Removable media                   S Sector data
    s Device status                     T SCSI sense data
    t Tape drive status                 U Device table
    u Staging queue

more (ctrl-f)
```

For the ASM/QFS-Standalone file system, Figure 26. shows the initial help screen.

Figure 26.   ASM/QFS-Standalone `samu`(1M) Initial Help Screen

```
Help information            page 1/5   samu 4.0-x Thu Oct 11
13:58:20

Displays:
    d Daemon trace controls          m Mass storage status
    f File systems                   C Memory
    h Help information               I Inode
    l License information            N File System Parameters

more (ctrl-f)
```

To move forward or backward from one screen to the next, enter the following key sequence:

- Press CTRL-f to page the display forward.
- Press CTRL-b to page the display backward to previous pages.

You can return to the help display at any time by pressing the h key.

Note: This manual does not describe the uppercase `samu`(1M) displays (`A`, `C`, `F`, `I`, `J`, `L`, `M`, `N`, `R`, `S`, `T`, and `U`) because they are designed to be used at a customer site only with the assistance of a member of the technical support staff.

Operator Displays

You can view the `samu`(1M) operator displays by pressing the key corresponding to each display. The lowercase keys `a` through `w` display operational information.

For displays that overflow the screen area, the word `more` appears on the bottom of the screen display, indicating that the display contains additional information. Figure 27. contains the word **more**, indicating that more information appears on subsequent screens.

Figure 27. `samu`(1M) Screen That Indicates More Text Can Be Obtained

```
xb54  54  exb8505  pt03  0  yes  2  0  on
lt55  55  dlt2000  pt02  1  yes  4  0  on  ml65
hp56  56  hpc1716  pt01  1  yes  3  0  on  hp70
hp57  57  hpc1716  pt01  1  yes  4  0  on  hp70
more
```

If `samu`(1M) prompts you to enter a device, enter its associated Equipment Ordinal. Equipment Ordinals for all devices are shown in the configuration display (`c`). To control all displays, use the control keys.

The following sections describe the operator displays. Examples are provided, and when necessary, displays are followed by a table describing the fields displayed.

(a) - Archiver Status Display

The archiver display shows the status of the archiver on a per-file-system basis.

Sample Display

Figure 28. shows activity and statistics for a single file system.

Figure 28. `samu`**(1M)** `a` **Display**

```
Archiver status samu 4.0.x Fri Jan 04 14:08:45

sam-archiverd:  Archiving files

sam-arfind:  samfs1 mounted at /sam1
Sleeping until Fri Jan 04 14:10:26 2002

sam-arcopy:  samfs1 arset0.2.9360 mo.opt06a
Copying file testdir0/filewh
```

Field Descriptions

To view the archiver detail display, enter `:a` *filesystem*. Table 52. shows the fields in the detail display.

Table 52. `samu`**(1M)** `a` **Display Field Descriptions**

Field	Description
`samfs1 mounted at`	Mount point.
`regular files`	Number of regular files and size.
`offline files`	Number of offline files and size.
`archdone files`	Number of `archdone` files and size. Indicates that the archiver has completed processing and can perform no further processing for `archdone` files. However, note that `archdone` files have not been archived.
`copy1`	Number of files and total size for archive copy 1.
`copy2`	Number of files and total size for archive copy 2.

Table 52. `samu`**(1M)** `a` **Display Field Descriptions (Continued)**

Field	Description
`copy3`	Number of files and total size for archive copy 3.
`copy4`	Number of files and total size for archive copy 4.
`Directories`	Number of directories and total size.
`sleeping until`	Indicates when archiver runs again.

(b) - Device Configuration Display

The configuration display shows your configuration's connectivity. To view the configuration display, press the `c` key.

Sample Display

Figure 29. shows the device configuration display.

Figure 29. `samu`**(1M)** `c` **Display**

```
Device configuration:     samu 4.0.x Thu Oct 11 13:10:23

ty  eq  state  device_name       fs family_set
ae  60  on     /dev/samst/c0t0u0 60 m160
at  61  on     /dev/rmt/0cbn     60 m160
at  62  on     /dev/rmt/1cbn     60 m160
at  63  on     /dev/rmt/3cbn     60 m160
at  64  on     /dev/rmt/4cbn     60 m160
hy  65  on     historian         65
```

Field Descriptions

Table 53. shows the field descriptions for this display.

Table 53. `samu`**(1M)** `c` **Display Field Descriptions**

Field	Description
`ty`	Device type.
`eq`	Equipment Ordinal of the device (unique number defined in the master configuration file).

Table 53. `samu`**(1M)** `c` **Display Field Descriptions**

Field	Description
`state`	Current operating state of the device. Valid device states are as follows: • `on`—The device is available for access. • `ro`—The device is available for read-only access. • `off`—The device is not available for access. • `down`—The device is available only for maintenance access. • `idle`—The device is not available for new connections. Operations in progress continue until completion.
`device_name`	Path to the device.
`fs`	Family set Equipment Ordinal.
`family_set`	Name of the storage family set or library to which the device belongs.

(c) - Daemon Trace Controls Display

The daemon trace controls display shows the events being traced as specified in the `defaults.conf` file. For more information on enabling trace files, see the `defaults.conf`(4) man page.

Sample Display

Figure 30. shows trace file information. It includes information on the daemons being traced, the paths to the trace files, the events being traced, and information on the size and age of the trace files.

Figure 30. `samu`**(1M)** `d` **Display**

```
Daemon trace controls       samu   4.0.5816 Fri Jan 18
10:42:02


sam-archiverd  /var/opt/SUNWsamfs/trace/archiver
               cust err misc files date module
               size    0    age 0

sam-catserverd /var/opt/SUNWsamfs/trace/catserver
               cust err fatal ipc misc proc queue ftp debug
date module
               size    0    age 0

ASMd        /var/opt/SUNWsamfs/trace/fsd
```

Figure 30. `samu`**(1M)** `d` **Display**

```
                cust err fatal ipc misc proc queue ftp debug
date module
                size    0    age 0

sam-ftpd        /var/opt/SUNWsamfs/trace/ftp
                cust err fatal ipc misc proc queue ftp debug
date module
                size    0    age 0

sam-recycler    /var/opt/SUNWsamfs/trace/recycler
                cust err fatal ipc misc proc queue ftp debug
date module
                size    0    age 0

sam-sharefsd    off



sam-stagerd     /var/opt/SUNWsamfs/trace/stager
                cust err misc proc files debug date module
                size    0    age 0
```

(d) - File Systems Display

The file systems display shows the components of your ASM/QFS-Standalone, ASM, or ASM-QFS file systems. To view the file systems display, press the `f` key.

Sample Display

Figure 31. shows the file systems display.

Figure 31. `samu`**(1M)** `f` **Display**

```
File systems                       samu   4.0.x Thu Oct 11
13:12:07

ty eq state      device_name      status high low mountpoint
server
ms  1    on            samfs1 m----2----d  80% 70%    /samfs1
md 11    on /dev/dsk/c2t5d0s5
md 12    on /dev/dsk/c2t6d0s5
```

Field Descriptions

Table 54. shows the field descriptions for this display.

Table 54. `samu`(1M) `f` Display Field Descriptions

Field	Description
`ty`	Device type.
`eq`	Equipment Ordinal of the device (unique number defined in the master configuration file).
`state`	Current operating state of the device. Valid device states are as follows: • `on`—The device is available for access. • `ro`—The device is available for read-only access. • `off`—The device is not available for access. • `down`—The device is available only for maintenance access. • `idle`—The device is not available for new operations. Operations in progress continue until completion.
`device_name`	File system name or path to the device.
`status`	Device status. For a description of status codes, see "Operator Display Status Codes" on page 247.
`high`	High disk usage threshold percentage.
`low`	Low disk usage threshold percentage.
`mountpoint`	Mount point of the file system.
`server`	Name of the host system upon which the file system is mounted.

(e) - License Display

The license display shows the licenses and expiration dates for ASM/QFS-Standalone, ASM, and ASM-QFS software. To view the configuration display, press the `l` key.

Sample Display

Figure 32. shows an example of a license display.

Figure 32. `samu`**(1M)** `l` **Display**

```
License Information samu 4.0.x Thu Oct 11 13:13:11

hostid = xxxxxxx

License never expires
Remote sam server feature enabled
Remote sam client feature enabled
Migration toolkit feature enabled
Fast file system feature enabled
Data base feature enabled
Direct media access feature enabled
Shared SAN filesystem support enabled
Segment feature enabled
Robot type ADIC 100 Library is present and licensed
    100 at slots present and licensed
Robot type DLT Tape Library is licensed
    100 lt slots licensed
Robot type IBM 3570 Changer is licensed
    100 i7 slots licensed
Robot type IBM 3584 Library is licensed
    100 li slots licensed
```

The sample display shows license information for an ASM file system. The license information is derived from the license keys in the following file:

```
/etc/opt/SUNWsamfs/LICENSE.4.0
```

The following information is displayed for the system:

- Expiration information
- Host ID
- ASM/QFS-Standalone, ASM, and ASM-QFS products and features enabled
- Equipment/media combinations

(f) - Mass-Storage Status Display

The mass-storage status display shows the status of mass-storage file systems and their member drives. To view the mass-storage status display, press the `m` key.

Sample Display

Figure 33. shows how member drives are indented one space and appear directly below the file system to which they belong.

Figure 33. `samu`**(1M)** `m` **Display**

```
Mass storage status                          samu 4.0.x Thu Oct 11
13:13:42

ty   eq  status       use  state  ord  capacity    free    ra
part  high  low
ms    1  m----2----d  21%  on          8.402G  6.644G  1024
16   80%  70%
 md  11               21%  on       0    4.251G  3.372G
 md  12               21%  on       1    4.151G  3.272G
```

Field Descriptions

Table 55. shows the field descriptions for this display.

Table 55. `samu`**(1M)** `m` **Display Field Descriptions**

Field	Description
`ty`	Device type.
`eq`	Equipment Ordinal of the mass-storage device.
`status`	Device status. For a description of status codes, see “Operator Display Status Codes” on page 247.
`use`	Percentage of disk space in use.
`state`	Current operating state of the mass-storage device.
`ord`	Ordinal number of the disk device within the storage family set.
`capacity`	Number of 1024-byte blocks of usable space on the disk.
`free`	Number of 1024-byte blocks of disk space available.
`ra`	Readahead size in kilobytes.
`part`	Partial stage size in kilobytes.
`high`	High disk usage threshold percentage.
`low`	Low disk usage threshold percentage.

(g) - Staging Status Display

The staging status display shows the status of the stager for all media. To view the staging status display, press the `n` key. To view the status for a specific device type, enter `:n` *media*, where *media* is the media type.

Sample Display

Figure 34. `samu`**(1M)** `n` **Display**

```
Staging status                    samu   4.0.x Thu Oct 11
13:14:23

Log output to:

Stage request: at.000004
Copying file /samfs1/testdir3/fileia

Stage request: at.000002
Copying file /samfs1/testdir1/fileei

Stage request: at.000003
Positioning for file /samfs1/testdir2/fileaa
```

(h) - Optical Disk Status Display

The optical disk status display shows the status of all optical disk drives configured within the ASM or ASM-QFS environment. To view the optical disk status display, enter `:o`.

Sample Display

Figure 35. `samu`**(1M)** `o` **Display**

```
Optical disk status            samu    4.0.x Thu Oct 11
13:15:40

ty  eq  status      act  use  state  vsn
mo  35  --l---wo-r    1  29%  ready  oper2
```

Field Descriptions

Table 56. shows the field descriptions for this display.

Table 56. `samu`**(1M)** `o` **Display Field Descriptions**

Field	Description
`ty`	Device type.
`eq`	Equipment Ordinal of the optical disk.
`status`	Device status. For a description of status codes, see “Operator Display Status Codes” on page 247.
`act`	Activity count.
`use`	Percentage of cartridge space used.

Table 56. `samu`**(1M)** `o` **Display Field Descriptions**

Field	Description
`state`	Current operating state of the optical disk. Valid device states are as follows: • `ready`—The device is on, and the disk is loaded in the transport; available for access. • `notrdy`—The device is on, but no disk is present in the transport. • `idle`—The device is not available for new connections. Operations in progress continue until completion. • `off`—The device is not available for access. • `down`—The device is available only for maintenance access.
`vsn`	Volume serial name assigned to the optical disk, or the keyword `nolabel` if the volume is not labeled.

(i) - Removable Media Load Requests Display

The removable media load requests display lists information on pending load requests for removable media. You can select either a specific type of media, such as DLT tape, or a family of media, such as tape. The priority display lists the priority in the preview queue rather than the user, and sorts the entries by priority.

Mount requests are displayed in three formats: both manual and robotic requests, manual requests only, or robotics requests only.

Enter only **:**`p` to display mount requests for all removable devices currently selected.

Enter **:**`p` *media_type* to display mount requests for devices of a given removable media type.

To select either the manual/robot display or the priority display, press the CTRL-k key sequence.

Sample Display 1

Figure 36. `samu`**(1M)** `p` **Display 1**

```
Removable media mount requests all  both  samu  4.0.x Fri Feb
9 11:21:42
                                                     count: 1
count   type  pid  user  rb   flags    wait count  vsn
   0    1t    473  root  40  Wb-f---  0:00         TAPE0
```

Sample Display 2

Figure 37. `samu`**(1M)** `p` **Display 2**

```
Removable media load requests all   priority samu   4.0.x Mon
Apr 26 21:44:27
License: License never expires.                        count: 3

index type pid  priority  rb  flags   wait  count  vsn
   0  i7   0        3007  70 ---f---  0:00        TAPE5
   2  i7   0           0  70 ---f---  0:00        TAPE1
  99  i7   1383   -49607  70 W--f---  0:06        TAPE14
```

Field Descriptions

Table 57. shows the field descriptions for this display.

Table 57. `samu`**(1M)** `p` **Display Field Descriptions**

Field	Description
`index`	Index number in the preview table.
`type`	Device type code assigned to the removable media.
`pid`	UNIX process identifier. A process identifier of 1 indicates NFS access.
`user`	Name assigned to the user requesting the load.
`priority`	Priority of the request.
`rb`	Equipment Ordinal of the robot in which the requested VSN resides.
`flags`	Flags for the device. See Table 58..
`wait`	The elapsed time since the mount request was received.
`count`	The number of requests for this VSN, if it is a stage.
`vsn`	Volume serial name of the volume.

Flags

Table 58. shows the flags.

Table 58. Flags Field for `samu`(1M) `p` Display

Field	Description
`W------`	Write access requested
`-b-----`	Entry is busy
`--C----`	Clear VSN requested
`---f---`	File system requested
`-----S-`	Flip side already mounted
`------s`	Stage request flag

(j) - Removable Media Status Display

The removable media status display enables you to monitor the activity on removable media devices such as tape drives. You can monitor either a specific type of device, such as video tape, or a family of devices such as all tape devices.

To view the status for all removable media devices, enter **:**`r`. To view the status for a specific device, enter **:**`r` *dt*, where *dt* is the device.

Sample Display

Figure 38. `samu`**(1M)** `r` **Display**

```
Removable media status: all    samu 4.0.x    Thu Oct 11
13:17:06

ty  eq  status     act  use  state  vsn
at  61  --l----o-r   1  73%  ready  000002
       0x541 blocks transferred
at  62  --l----o-r   1  70%  ready  000004
       0x7da blocks transferred
at  63  --l----o-r   1  90%  ready  000003
       0x2a0 blocks transferred
at  64  --l------r   0  54%  ready  000001
       idle
```

Field Descriptions

Table 59. shows the field descriptions for this display.

Table 59. `samu`**(1M)** `r` **Display Field Descriptions**

Field	Description
`ty`	Device type.
`eq`	Equipment Ordinal of the drive.
`status`	Device status. For a description of status codes, see “Operator Display Status Codes” on page 247.
`act`	Activity count.
`use`	Percentage of cartridge space used (optical disk only).
`state`	Current operating state of the removable media. Valid device states are as follows: • `ready`—The device is on, and the disk or tape is loaded in the transport; available for access. • `notrdy`—The device is on, but no disk or tape is present in the transport. • `idle`—The device is not available for new connections. Operations in progress continue until completion. • `off`—The device is not available for access. • `down`—The device is available only for maintenance access.
`vsn`	Volume serial name assigned to the volume, or the keyword `nolabel` if the volume is not labeled. Blank if no volume is present in the transport, or device is off.

(k) - Device Status Display

The device status display shows the status for all devices configured within the ASM or ASM-QFS environment. To view the device status summary display, enter `:s`.

Sample Display

Figure 39. `samu`**(1M)** `s` **Display**

```
Device status              samu  4.0.x     Thu Oct 11
13:18:18

ty  eq  state  device_name       fs  status      pos
ae  60  on     /dev/samst/c0t0u0  60  m--------r
     move complete
at  61  on     /dev/rmt/0cbn      60  --l----o-r
     0x70d blocks transferred
at  62  on     /dev/rmt/1cbn      60  --l----o-r
     0x986 blocks transferred
at  63  on     /dev/rmt/3cbn      60  --l----o-r
     0x46d blocks transferred
at  64  on     /dev/rmt/4cbn      60  --l------r
     idle
hy  65  on     historian          65  ----------
```

Field Descriptions

Table 60. shows the field descriptions for this display.

Table 60. `samu`**(1M)** `s` **Display Field Descriptions**

Field	Description
`ty`	Device type.
`eq`	Equipment ordinal of the device.
`state`	Current operating state of the device.
`device_name`	Path to the device. For file system devices, this is the file system name.
`fs`	Equipment Ordinal of the family, set to which the device belongs.
`status`	Device status. For a description of status codes, see “Operator Display Status Codes” on page 247.
`pos`	Device position.

(l) - Tape Drive Status Display

The tape drive status display shows the status of all tape drives configured within the ASM or ASM-QFS environment. To view the tape status display, press the t key.

Sample Display

Figure 40. samu(1M) t Display

```
Tape drive status              samu   4.0.x Thu Oct 11
13:18:48

ty  eq  status      act  use  state  vsn
at  61  --l----o-r    1  73%  ready  000002
       0x7b7 blocks transferred
at  62  --l----o-r    1  70%  ready  000004
       0xa35 blocks transferred
at  63  --l----o-r    1  90%  ready  000003
       0x518 blocks transferred
at  64  --l----o-r    1  54%  ready  000001
       0x20 blocks transferred
```

Field Descriptions

Table 61. shows the field descriptions for this display.

Table 61. samu(1M) t Display Field Descriptions

Field	Description
ty	Device type.
eq	Equipment Ordinal of the drive.
status	Device status. For a description of status codes, see “Operator Display Status Codes” on page 247.
act	Activity count.
use	Percentage of cartridge space used (optical disk only).

Table 61. samu(1M) t Display Field Descriptions

Field	Description
state	Current operating state of the removable media. Valid device states are as follows: • ready—The device is on and the disk or tape is loaded in the transport; available for access. • notrdy—The device is on but no disk or tape is present in the transport. • idle—The device is not available for new connections. Operations in progress continue until completion. • off—The device is not available for access. • down—The device is available only for maintenance access.
vsn	Volume serial name assigned to the volume, or the keyword nolabel if volume is not labeled. Blank if no volume is present in the transport, or device is off.

(m) - Staging Queue Display

The samu(1M) utility's u display lists all files in the staging queue. To select this display, type u. Press the CTRL-k key sequence to list the file path name on the second line of each entry.

Sample Display

Figure 41. `samu`**(1M)** `u` **Display**

```
Staging queue by media type: all samu 4.0.x Thu Oct 11
13:19:34
volumes 2 files 827

ty    length  fseq   ino  position  offset  vsn

at    1.674M     1  2513     389d4  7e70b   000004
at    1.875M     1  2640     389d4  7f470   000004
at    1.643M     1  1536     389d4  80372   000004
at    1.063M     1   248     389d4  81099   000004
at  562.037k     1   595     389d4  8191b   000004
at    1.000M     1   142     389d4  81d81   000004
at    1.264M     1   442     389d4  82582   000004
at  599.014k     1  2237     389d4  82fa0   000004
at  816.685k     1  2435     389d4  83450   000004
at    1.429M     1  2701     389d4  83ab3   000004
at    1.752M     1   439     389d4  84623   000004
at    1.089M     1   565     389d4  85428   000004
at  975.326k     1   121     389d4  85ce1   000004
at    1.014M     1    28     389d4  86481   000004
at  683.581k     1   419     389d4  86c9f   000004
at    1.562M     1  1608     389d4  871f8   000004
   more
```

Field Descriptions

Table 62. shows the field descriptions for this display.

Table 62. `samu`**(1M)** `u` **Display Field Descriptions**

Field	Description
`ty`	Device type.
`length`	File length.
`fseq`	File system equipment number.
`ino`	The inode number.
`position`	The position (in decimal format) of the archive file on the specific medium.
`offset`	Offset of the archive file on the specific medium.
`vsn`	Volume serial name of the volume.

(n) - Robot Catalog Display

The robot catalog display shows the location and VSN of all disks or tapes currently cataloged in the robot. To view the library VSN catalog display, press the `v` key. If the operator utility prompts for a robot name, enter either the device name or an Equipment Ordinal. A null entry displays the last library shown. For a list of all device names and Equipment Ordinals, view the configuration display by pressing the `c` key.

The CTRL-k key sequence changes the sorting key for this display. The CTRL-i key sequence changes to a two-line display that shows the times and barcodes. Pressing the CTRL-i key sequence a second time displays volume reservation information on the second line.

Sample Display

Figure 42. `samu`**(1M)** `v` **Display**

```
Robot VSN catalog by slot : eq 60 samu 4.0.x Thu Oct 11
13:20:04
                                                   count 32
slot       access time  count  use  flags          ty  vsn

   0  none                70    0%  -il-oCb-----   at  CLN005
   1  2001/10/11 08:31    10   90%  -il---b-----   at  000003
   2  2001/10/11 13:07    17   73%  -il---b-----   at  000002
   3  2001/10/11 12:48    16   70%  -il---b-----   at  000004
   4  2001/10/11 12:55    30   54%  -il---b-----   at  000001
   5  none                 0    0%  -il-o-b-----   at  000005
   6  none                 0    0%  -il-o-b-----   at  000044
   7
  13  2001/10/11 13:05    61    0%  -il-o-b-----   at  000033
```

Field Descriptions

Table 63. shows the field descriptions for this display.

Table 63. `samu`**(1M)** `v` **Display Field Descriptions**

Field	Description
`Robot VSN catalog`	Name of the specified robot and time the display refreshed.
`count`	Number of slots in library.
`slot`	Slot number within the specified library.
`access time`	Time the optical disk was last accessed.
`count`	Number of accesses to this volume since the last audit operation.

Table 63. samu(1M) v Display Field Descriptions (Continued)

Field	Description
use	Percentage of space used for the volume.
flags	Flags for the device. See Table 64. for information on the flags.
ty	Device type.
vsn	Volume serial name of the volume.

Flags

In some cases, more than one flag can occur in a field, and one flag overrides the other. Table 64. shows the flags from the flags field from Table 63..

Table 64. Flags Field for samu(1M) v Display

Flags	Description
A----------	Volume needs audit.
-i---------	Slot in use.
--l--------	Labeled. Overrides N.
--N--------	Unlabeled. This volume is foreign to the ASM or ASM-QFS environment.
---E-------	Media error. Set when the ASM or ASM-QFS software detects a write error on a cartridge.
----o------	Slot occupied.
-----C-----	Volume is a cleaning tape. Overrides p.
-----p-----	Priority VSN.
------b----	Barcode detected.
-------W---	Write protect. Set when the physical write protection mechanism is enabled on a cartridge.
--------R---	Read only.
---------c--	Recycle.

Table 64. Flags Field for `samu`(1M) `v` Display

Flags	Description
`----------d-`	Duplicate VSN. Overrides `U`.
`----------U-`	Volume unavailable.
`-----------X`	Export slot.

(o) - Pending Stage Queue

The pending stage queue display shows queued stage requests for which the volumes have not yet been loaded. Press the CTRL-k key sequence to list the path name on the second line of each entry.

Sample Display

Figure 43. `samu`**(1M)** `w` **Display**

```
Pending stage queue by media type: all     samu     4.0.x Thu
Oct 11 13:20:27
                                                        volumes
1 files 13

ty     length  fseq  ino  position  offset  vsn

at     1.383M     1   42     3a786    271b  000002
at     1.479M     1   56     3a786    5139  000002
at  1018.406k     1   60     3a786    6550  000002
at     1.000M     1   65     3a786    7475  000002
at     1.528M     1   80     3a786    99be  000002
at     1.763M     1   92     3a786    ce57  000002
at     1.749M     1  123     3a786   11ece  000002
at   556.559k     1  157     3a786   1532f  000002
at   658.970k     1  186     3a786   17705  000002
at   863.380k     1  251     3a786   1dd58  000002
at     1.268M     1  281     3a786   1f2b7  000002
at     1.797M     1  324     3a786   23dfa  000002
at     1.144M     1  401     3a786   2bb6d  000002
```

Field Descriptions

Table 65. shows the field descriptions for this display.

Table 65. `samu`**(1M)** `w` **Display Field Descriptions**

Field	Description
`ty`	Device type.
`length`	File length.
`fseq`	File system Equipment Ordinal.
`ino`	The inode number.
`position`	The position (in decimal format) of the archive file on the specific medium.
`offset`	Offset of the archive file on the specific medium.
`vsn`	Volume serial name of the volume.

Operator Display Status Codes

The operator displays have different status codes for the removable media device displays versus the file system displays. The following sections describe these displays.

Removable Media Device Display Status Codes

The c, o, r, s, and t operator displays show status codes for removable media devices. Status codes are displayed in a 10-position format, reading from left (position 1) to right (position 10).

The status codes in this section do not apply to the samu(1M) f, m, and v displays. For information on the status codes for the f and m displays, see “File System Display Status Codes” on page 249. For information on the status codes for the v display, see “(n) - Robot Catalog Display” on page 244.

Table 66. defines the valid status codes for each position.

Table 66. Removable Media Device Display Status Codes

Status Bit	Meaning for a Device
`s---------`	Media is being scanned.
`M---------`	Maintenance mode.
`-E--------`	Device received an unrecoverable error in scanning.
`-a--------`	Device is in audit mode.
`--l-------`	Media has a label.
`--N-------`	Foreign media.
`---I------`	Waiting for device to idle.
`---A------`	Needs operator attention.
`----C-----`	Needs cleaning.
`----U-----`	Unload has been requested.
`-----R----`	Device is reserved.
`------w---`	A process is writing on the media.
`-------o--`	Device is open.

Table 66. Removable Media Device Display Status Codes (Continued)

Status Bit	Meaning for a Device
`--------P-`	Device is positioning (tape only).
`--------F-`	For robots, all storage slots occupied. For tape and magneto optical drives, media is full.
`---------R`	Device is ready and the media is read-only.
`---------r`	Device is spun up and ready.
`---------p`	Device is present.
`---------W`	Device is write protected.

File System Display Status Codes

The `f` and `m` operator displays show status codes for file systems. Status codes are displayed in an 11-position format, reading from left (position 1) to right (position 11).

The status codes in this section do not apply to the `samu`(1M) `c`, `o`, `r`, `s`, `t`, or `v` displays. For information on the status codes for the c, o, r, s, and t displays, see “Removable Media Device Display Status Codes” on page 248. For information on the status codes for the v display, see “(n) - Robot Catalog Display” on page 244.

Table 67. defines the valid status codes for each position.

Table 67. File System Display Status Codes

Status Bit	Meaning for a File System
`m----------`	File system is currently mounted.
`M----------`	File system is being mounted.
`-u---------`	File system is being unmounted.
`--A--------`	File system data is being archived.
`---R-------`	File system data is being released.
`----S------`	File system data is being staged.
`-----1-----`	ASM or ASM-QFS file system version 1.
`-----2-----`	ASM or ASM-QFS file system version 2.
`------C----`	ASM/QFS-Standalone shared file system.

Table 67. File System Display Status Codes

Status Bit	Meaning for a File System
`-------W---`	Single writer.
`--------R--`	Multireader.
`---------r-`	`mr` devices.
`----------d`	`md` devices.

Operator Display Device States

The `c`, `m`, `o`, `r`, `s`, and `t` operator displays show device state codes. These codes represent the current access state for the device.

You can use `samu`(1M) to change the state of a device. The following examples show a typical progression to change a drive's state from `down` to `on` and from `on` to `down`:

- **Example 1**. The following progression can be used to change a device state from `down` to `on`:

  down -> off -> [unavail] -> on

  In this progression, the brackets indicate that it is not necessary to pass through the `unavail` state.

- **Example 2**. The following progression can be used to change a device state from `on` to `down`:

  on -> [idle] -> [unavail] -> off -> down

  In this progression, the brackets indicate that it is not necessary to pass through the `idle` or `unavail` states.

Table 68. defines the valid state codes.

Table 68. Operator Display Device States

Device State	Description
`on`	The device is available for access. For certain displays, this state may be superseded by the states `ready` or `notrdy`.
`ro`	The device is available for read-only access. Like `on`, this state can be superseded for certain displays by `ready` or `notrdy`.

Table 68. Operator Display Device States (Continued)

Device State	Description
off	The device is not available for access. For tape and optical disk drives, possible reasons for the device to be in the off state include the following: • Cleaning was requested, but no cleaning cartridge was found in the automated library. • The cleaning cartridge cannot be loaded or unloaded from the drive. • Initialization found the drive status to be full, and attempts to clear the drive failed. • The system was unable to clear a cartridge from a drive. • Opening the drive for I/O failed during spin-up. • An error other than NOT READY was received when spinning the drive down for unloading. • Opening the standard tape driver on the drive failed during spin up.
down	The device is available for maintenance access only.
idle	The device is not available for new connections. Operations in progress continue until completion.
ready	The device is on and the disk or tape loaded in the transport is available for access.
notrdy	The device is on, but no disk or tape is present in the transport.
unavail	The device is unavailable for access and cannot be used for automatic ASM or ASM-QFS operations. You can continue to use the load and unload commands for placing and removing media from the device while it is in the unavail state.

Operator Commands

This section describes the following types of operator commands:

- “Archiver Commands” on page 252
- “Device Commands” on page 253
- “Display Control Commands” on page 254
- “File System Commands” on page 256
- “Robot Commands” on page 257
- “Miscellaneous Commands” on page 258

Note: If you want to enter any operator commands from the Sun Solaris operating environment (OE) command line, you must use them as arguments to the `samcmd`(1M) command. For more information on the `samcmd`(1M) command, see the `samcmd`(1M) man page.

Each `samu`(1M) command is prefaced with a colon (:) when it is entered to designate that a command line command is being entered and not a series of hot keys.

Archiver Commands

Table 69. shows the archiver commands and their actions.

Table 69. Archiver Command Actions

Command	Action
`aridle`	Stops all archiving at the next convenient point. For example, at the end of the current `tar`(1) file for `sam-arcopy` operations. This command can be used, for example, to stop all archiving activity for all file systems prior to unmounting the file systems.
`arrestart`	Interrupts the archiver and restarts the archiver. This action occurs regardless of the state of the archiver. Therefore, `arrestart` should be used with caution. Some copy operations to archive media might not complete and must be repeated. This wastes space on the media.
arrun	Causes the archiver to begin archiving. This command overrides any existing global `wait` command in the `archiver.cmd` file.
`arstop`	Stops all archiving immediately.

The formats for the archiver commands are as follows:

```
:aridle [ dk | rm | fs.fsname ]
:arrestart
:arrun [ dk | rm | fs.fsname ]
:arstop [ dk | rm | fs.fsname ]
```

The arguments to these commands are optional. If no arguments are specified, all file systems are affected. If arguments are specified, the command takes action based on the type of archive file specified (`dk` or `rm`) and the file system specified. Table 70. shows the archiver command arguments.

Table 70. Archiver Command Arguments

Argument	Description
`dk`	Specifies that this command pertains to disk archive files.
`rm`	Specifies that this command pertains to removable media files.
`fs.`*fsname*	Specifies that this command pertains to a specific file system. Enter a file system name for *fsname*.

Device Commands

Table 71. shows the device commands and their actions.

Table 71. Device Command Actions

Command	Action
devlog	Sets device-logging options.
down	Terminates operation on device *eq*.
idle	Restricts access to device *eq* by preventing new connections to the device. Existing operations continue until completion.
off	Logically turns off device *eq*.
on	Logically turns on device *eq*.
unavail	Selects device *eq* and makes it unavailable for use with the ASM or ASM-QFS file system.
`unload`	Unloads the mounted media for the specified removable media device *eq*. For magazine devices, the `unload` command unloads the mounted cartridge and ejects the magazine.

The formats for the device control commands are as follows:

```
:devlog eq [ option ...]
:down eq
:idle eq
:off eq
:on eq
:unavail eq
:unload eq
```

Table 72. shows the device command arguments.

Table 72. Device Command Arguments

Argument	Description
eq	The Equipment Ordinal of a device in the `mcf` file.
option	Zero or more event types. Possible event types are as follows: `all`, `date`, `default`, `detail`, `err`, `event`, `label`, `mig`, `module`, `msg`, `none`, `retry`, `stage`, `syserr`, and `time`. For information on these options, see the `defaults.conf`(4) man page.

Display Control Commands

Table 73. shows the display control commands and their actions.

Table 73. Display Control Command Actions

Command	Action
`:a` [*filesystem*]	Displays the archiver status.
`:n` [*media*]	Selects the media type for the removable media I/O activity display.
`:p` [*media*]	Selects the media type for the mount requests display.
`:q`	Causes the `samu` operator utility to exit.
`:r` [*media*]	Selects the device type for the removable media status display.
`:refresh` *i*	Sets the time interval for refreshing the display window and enables display refreshing. The CTRL-r key sequence toggles display refreshing on and off.
`:u` [*media*]	Displays the stage queue. This pertains to currently mounted volumes.
`:v` [*eq*]	Selects the library VSN catalog for display. To view the VSNs in the historian catalog, enter the keyword `historian` in place of *eq*.
`:w` [*media*]	Displays the prestage queue. This pertains to volumes that are not yet mounted.

The formats for the display control commands are as follows:

```
:a [ filesystem ]
:n [ media ]
:p [ media ]
:q
:r [ media ]
:refresh i
:u [ media ]
:v [ eq ]
:w [ media ]
```

The brackets around the arguments to these commands show that in many cases, the arguments are optional. The arguments to many of these commands narrow the `samu`(1M) display output to a specific file system, media type, or Equipment Ordinal. If no arguments are specified, the command displays information for all file systems, media types, and Equipment Ordinals that are currently selected or configured.

Table 74. shows the display control command arguments.

Table 74. Display Control Command Arguments

Argument	Description
filesystem	Specifies the name of a ASM or ASM-QFS file system. If the *filesystem* argument is specified, the archiver status display shows the number of regular files; the number of offline files; the number of archived files; the number of archive copies and directories; file systems; mount points; inode activity; and interval. If the *filesystem* argument is not specified, the archiver status display shows the name of the file system and mount point, scans for inode activity, and lists the next time the archiver will scan the file system.
media	Specifies a media type. For a list of supported media types, see the `mcf`(4) man page. The keyword `all` can also be specified to represent all media types or removable media devices.
eq	The Equipment Ordinal of a device in the `mcf` file.
i	The time interval in seconds.

File System Commands

The :meta_timeo eq interval Command

The `metatimeo` command sets the ASM/QFS-Standalone shared file system metadata cache time out value.

For *eq*, specify the Equipment Ordinal of the file system.

For *interval*, specify an interval in seconds. The default *interval* is 15. After this interval expires, the client host systems obtain a new copy of the metadata information from the metadata server host.

The :notrace eq Command

The `notrace` command disables tracing.

For *eq*, specify the Equipment Ordinal of the file system.

The :partial eq size Command

The `partial` command sets the number of kilobytes to leave online after release of the file.

For *eq*, specify the Equipment Ordinal for the file system.

For *size*, specify the number of kilobytes to leave online. The default *size* is 16.

The :readahead eq contig Command

The `readahead` command specifies the maximum number of bytes that can be read ahead by the file system.

For *eq*, specify the Equipment Ordinal for the file system.

For *contig*, specify units of 1-kilobyte blocks. This must be an integer such that 1 < *contig* < 8192. The contig specified is truncated to a multiple of 8 kilobytes. The default *contig* is 8 (131072 bytes).

For example, the following command sets the maximum contiguous block size to 262,144 bytes for the file system defined as Equipment Ordinal `3`:

```
:readahead 3 256
```

This value can also be configured in the `samfs.cmd` file by specifying the readahead directive. For more information, see the `samfs.cmd`(4) man page.

The :thresh eq high low Command

The `thresh` command sets the high and low thresholds for a file system to control file archiving.

For *eq*, specify the Equipment Ordinal of the storage family set.

For *high*, specify the high threshold.

For *low*, specify the low threshold.

For example, the following command sets the high threshold to 50 percent and the low threshold to 40 percent for the storage family set whose file system Equipment Ordinal is 10:

```
:thresh 10 50 40
```

The :trace eq Command

The `trace` command enables tracing for a file system.

For *eq*, specify the Equipment Ordinal of a file system.

The :writebehind eq contig Command

The `writebehind` command specifies the maximum number of bytes that can be written behind by a file system.

For *eq*, specify the Equipment Ordinal for a file system.

For *contig*, specify units of 1-kilobyte blocks. This must be an integer such that 1 < *contig* < 8192. The default *contig* is 8 (131072 bytes).

For example, the following command sets the maximum contiguous block size to 262,144 bytes for the file system defined as Equipment Ordinal `50`:

```
:writebehind 50 256
```

This value can also be configured in the `samfs.cmd` file by specifying the `writebehind` directive. For more information, see the `samfs.cmd`(4) man page.

Robot Commands

The :audit [-e] eq [:slot [:side] Commands

The `audit` command causes the specified robotic device to mount each volume, read the VSN, and rebuild the library catalog.

For *eq*, specify the Equipment Ordinal of a robotic device.

The :export eq:slot and :export mt.vsn Commands

The `export` command causes the specified robotic device to export a volume to the mail slot. The volume is identified by its slot position within the robot.

- If exporting by Equipment Ordinal and slot number, the specified robotic device loads the volume into a drive. For *eq*, specify the Equipment Ordinal or device name. For *slot*, specify the slot number containing the volume you want to load.
- If exporting by logical identifier, the specified robotic device to mounts a labeled volume in to a drive. For *mt*, specify the media type; for information on valid media types, see the `mcf`(4) man page. For *vsn*, specify the volume to mount.

The :import eq Command

The `import` command causes the specified robotic device to allow you to add a cartridge. For *eq*, specify the Equipment Ordinal of the robotic device.

The :load eq:slot [:side] and :load mt.vsn Commands

The `load` command enables you to load by either a physical or a logical identifier, as follows:

- If loading by Equipment Ordinal and slot number, the specified robotic device loads the volume into a drive. For *eq*, specify the Equipment Ordinal or device name. For *slot*, specify the slot number containing the volume you want to load.
- If loading by logical identifier, the specified robotic device to load mounts a labeled volume into a drive. For *mt*, specify the media type; for information on valid media types, see the `mcf`(4) man page. For *vsn*, specify the volume to mount.

Miscellaneous Commands

The :clear vsn [index] Command

The `clear` command clears the specified VSN from the removable media mount requests display (see “(i) - Removable Media Load Requests Display” on page 236). Any process waiting for the VSN mount is aborted. If *index* is specified, *index* is the decimal ordinal of the VSN in the removable media display.

The :dtrace Commands

The `dtrace` commands are as follows:

- `:dtrace` *daemon_name* `on`
- `:dtrace` *daemon_name* `off`
- `:dtrace` *daemon_name*`.`*variable value*

The `dtrace` commands specify various tracing options. Table 75. shows the tracing control command arguments.

Table 75. Tracing Command Arguments

Argument	Description
daemon_name	Specify the keyword `all` or a process name. If the keyword `all` is specified, the tracing command affects all daemons. If one of the following process names is specified, the tracing command affects that process only: `sam-archiverd`, `sam-catserverd`, `ASMd`, `sam-ftpd`, `sam-recycler`, `sam-sharefsd`, and `sam-stagerd`. One of the keywords `on` or `off` can be specified after a process name. If `on` or `off` are specified, tracing is turned off or on for all processes specified.
variable value	Many different *variable* and *value* arguments can be specified. The `defaults.conf`(4) man page contains comprehensive information on these arguments. Specify one of the following *variable* and *value* combinations: • `file` *value*. For *value*, specify the name of a file to which trace files can be written. This can be a full path name. • `options` *value*. For *value*, specify a space-separated list of trace options. • `age` *value*. For *age*, specify the trace file rotation age. • `size` *value*. For *value*, specify the size of the trace file at which rotation will begin.

The :mount mntpt Command

The `mount` command selects an ASM/QFS-Standalone, ASM, or ASM-QFS file system.

The :open eq Command

The `open` command enables access to the specified disk device. This command must be issued before you can use the `read` command, disk sector display (`S`), or file label display (`F`). *eq* is the Equipment Ordinal.

The :read addr Command

The `read` command reads the specified sector from the currently opened disk device. You must open the device before it can be read. For *addr*, specify the hexadecimal sector address.

The :snap [filename] Command

The snap command sends a snapshot of a display window to *filename*, which is the name of a file to receive the display information.

To aid in problem reporting, you can take a snapshot of all the samu(1M) utility's displays. Each new snapshot is appended to the snapshots file. The default file is snapshots in the current working directory. The file can be printed, examined using vi(1), or faxed to StorageTek customer support staff.

The :! shell_command Command

The ! command allows you to run a shell command without leaving the samu(1M) operator utility.

Upgrading Your Environment 10

This chapter describes how to upgrade the hardware within your existing ASM or ASM-QFS environment. The following topics are presented:

Certain other types of operations and upgrades also need to be performed within a ASM or ASM-QFS environment. The following publications describe these other types of upgrades:

- The *ASM, ASM-QFS, and ASM/QFS-Standalone File System Administrator's Guide* describes the following types of operations and upgrades:
  - How to initialize a file system
  - How to initialize or reinitialize an `mcf` or `defaults.conf` file
  - How to mount a file system
  - How to unmount a file system
  - How to check file system integrity
  - How to repair a file system
  - How to preserve information for an upgrade
  - How to prepare for a hardware upgrade
  - How to add disk cache to a file system
  - How to replace disks in a file system
  - How to upgrade a host system
  - How to upgrade your Sun Solaris operating environment (OE) in an ASM/ QFS-Standalone, ASM, or ASM-QFS environment
  - How to upgrade your Sun Solaris operating environment (OE) in an ASM/ QFS-Standalone environment
- The *ASM, ASM-QFS, and ASM/QFS-Standalone Installation and Configuration Guide* describes upgrading ASM/QFS-Standalone, ASM, and ASM-QFS software.

■ To Add Slots in an Automated Library

The number of cartridge slots managed by the ASM or ASM-QFS system is controlled by a license key. To increase the number of slots, follow the steps in this section.

To Add Slots In a Library

1. Determine whether you need to obtain a new set of license keys through your authorized service provider (ASP) or, if an ASP is not assigned to your account, through StorageTek. (Optional)

2. Replace the existing license keys with the new license keys. (Optional)

   Perform this step if you obtained new license keys.

License keys start in column 1 of the following file:

```
/etc/opt/SUNWsamfs/LICENSE.4.0
```

No other keywords, host IDs, or other information can appear.

3. Issue the `samd config` command to enable the ASM or ASM-QFS software to recognize the new license keys. (Optional)

   Perform this step if you obtained new license keys.

   For example:

```
# samd config
```

4. Unload the library catalog.

   You can use the `samu`(1M) utility's `:unload` command, as follows:

```
: unload eq
```

   The *eq* argument identifies the Equipment Ordinal of the automated library as defined in the `mcf` file. This command moves the library catalog entries into the historian catalog and preserves the catalog information for each cartridge.

   After the `:unload` command is entered, in `samu`(1M), the automated library's `v` display empties, and the historian's `v` display fills up with the VSNs that used to be in the automated library.

5. Bring down the ASM or ASM-QFS system.

For information on how to perform this step, see “Basic Operations” on page 13.

6. Power down the host system and the library according to the manufacturer’s suggested procedure.
7. Have the library hardware engineer add slots to the automated library.
8. Power on the system using your normal start-up procedure.
9. Start the ASM or ASM-QFS system.

For information on how to perform this step, see “Basic Operations” on page 13. The new license information appears in the `samu`(1M) utility’s `l` display.

■ To Upgrade or Replace a Library

Prior to disconnecting and installing a different automated library, prepare for the upgrade as described in the *ASM, ASM-QFS, and ASM/QFS-Standalone File System Administrator’s Guide* in the section called "Preparing for Hardware Upgrades".

To Replace or Upgrade a Library

1. Unload the library catalog using the `samu`(1M) utility’s `:unload` command.

For example:

```
:unload eq
```

The *eq* argument identifies the Equipment Ordinal of the automated library as defined in the `mcf` file. This command moves the library catalog entries into the historian catalog and preserves the catalog information for each cartridge.

After the `:unload` command is entered, the automated library’s `v` display empties, and the historian’s `v` display fills up with the VSNs that used to be in the automated library.

2. Update the `/etc/opt/SUNWsamfs/inquiry.conf` file. (Optional)

The new library should be identified in this file by the vendor, the automated library model, and a ASM or ASM-QFS internal name.

For example, the released `inquiry.conf` file includes the following line:

```
"HP",   "C1710T",  "hpoplib"   # HP optical library
```

This line indicates that if the system detects a SCSI device made by vendor `HP` of model `C1710T`, the system drives it as an `hpoplib`. The first two fields (vendor/product) are returned from the hardware devices. The last field, `hpoplib`, is a name that the system uses internally to determine how to communicate with the device. If the `inquiry.conf` file needs to be changed, the change does not become effective until the `sam-initd` daemon is restarted.

3. Save the current `/etc/vfstab` file as `/etc/vfstab.cur`.
4. Edit the `/etc/vfstab` file.

   Change any ASM or ASM-QFS mounts from `yes` to `no`.
5. Save the `/etc/opt/SUNWsamfs/archiver.cmd` file as `archiver.cmd.cur`.
6. Edit the `/etc/opt/SUNWsamfs/archiver.cmd` file

   Add a `wait` directive as the first line.
7. Power off the host system and peripherals using the manufacturer's suggested procedure.
8. Disconnect the automated library.
9. Attach the connecting cable to the new automated library.
10. Power on the peripherals and the host system using the suggested power-on sequence.
11. Ensure that the host system identifies the new automated library.

Enter the following command:

```
> probe-scsi-all
```

The new automated library and its drives must be displayed prior to proceeding. If these devices are not identified, the automated library and its drives probably have a connection problem.

12. Boot the system.

Enter the following command to boot with the new configuration:

```
> boot -rv
```

13. If the target numbers of the drives or automated library changed, or if the ordering or number of the drives in the automated library changed, modify the `/etc/opt/SUNWsamfs/mcf` file to reflect the new configuration. (Optional)

This is similar to an initial installation as described in the *ASM, ASM-QFS, and ASM/QFS-Standalone Installation and Configuration Guide*.

14. Create new `/dev/samst` entries. (Optional)

    Perform this step if you are adding new equipment. Enter the following command:

```
# samdev
```

15. Initialize the ASM or ASM-QFS system.

    You can perform this step by either mounting a file system or by entering the following command:

```
# samd start
```

When the system initializes, it recognizes that the number of slots in the automated library has changed. The system runs a full audit on the automated library to update the library catalog. A full audit must be completed before archiving is resumed.

If there are problems in the audit, the most likely reason is that the ordering of the drives in the automated library does not match the ordering in the `/etc/opt/SUNWsamfs/mcf` file. Remember, that drives have two attributes: the SCSI target ID and the position in the automated library. Both of these attributes must be correct both before and after the upgrade.

If the audit completes without problems, proceed to the next step.

16. Replace the `/etc/vfstab` and `/etc/opt/SUNWsamfs/archiver.cmd` files with the pre-upgrade versions.

    Use the saved `/etc/vfstab.cur` and `/etc/opt/SUNWsamfs/archiver.cur` files, respectively.

17. Reboot the system to ensure that no errors exist in the configuration.

The automated library calls the drives by position number. When the system wants to load a cartridge into a drive, it must, for example, send a command to the automated library to load a cartridge from slot 123 into drive 3.

Drive 3 might be SCSI target 6 based on the third mcf entry. The system knows it is drive 3 because it is the third drive entry in the mcf file. The automated library knows that it is drive 3 because of the physical location it occupies in the automated library.

After the automated library has been requested to load the cartridge into the drive, the system tests the drive for unit ready status. Here, the system uses the SCSI target ID as defined in the `/dev/samst/`*scsi-target* entry in the `mcf` file. Therefore, it is important that the entry match the drive that was just loaded with the cartridge.

There is no good way to determine this information. Usually, the manufacturer ships the automated library set up with ascending SCSI IDs on the drives, but there is no guarantee of this. One way to determine this is to use the `samu`(1M) utility's `:load` command to load a cartridge, and then watch the `samu`(1M) utility's `s` display to see which drive shows the `r`, rather than the `p`, in the status flags of the `t` display.

■ To Upgrade DLT Tape Drives

To take advantage of the higher density and faster tape technology, it is often desirable to upgrade DLT tape drives in an automated library or a standalone tape drive. For example, you can move from DLT 4000 drives to DLT 7000 drives.

In a ASM or ASM-QFS environment, this is a matter of adding the new drive, rebooting the new configuration, and updating the `mcf` file as necessary, prior to starting the ASM environment. In addition, if you are upgrading the number of slots, you need to contact your authorized service provider (ASP) or StorageTek because you might need an upgraded license.

Note the following restrictions and general information before upgrading drives:

- ASM and ASM-QFS environments do not support mixed DLT tape drives within the same direct-attached automated library. For example, an ASM system cannot differentiate between a DLT 4000 tape drive and a DLT 7000 tape drive in the same automated library. Therefore, you should plan on replacing all the DLT drives with the new drives at the same time.
- The lower-density tapes can coexist with higher-density tapes and tape drives. You can continue to read and write to the lower-capacity tapes using a higher-density drive.
- To take full advantage of the higher-density DLT tapes, you may want to recycle existing files and migrate them to a higher-density tape. You can accomplish this by marking all the lower-density tapes as read-only, and then marking these tapes to be recycled. For information on recycling tape, see "Recycling" on page 177.
- As each tape is labeled, the density of the tape is acknowledged and recorded in the library catalog.

To Upgrade Tape Drives

1. Decide whether or not your current dump files are sufficient. (Optional)

If you decide that they are not, perform a `samfsdump`(1M) on your file systems prior to proceeding to the next step.

2. Update the `/kernel/drv/st.conf` file to identify the new drives.

The tape drives are identified in this file by the vendor, the tape model, and a ASM or ASM-QFS internal name. For example, the released `st.conf` file contains the following line:

```
"QUANTUM DLT7000", "DLT 7000 tape drive",   "dlt7-tape"
```

An example file is provided in `/opt/SUNWsamfs/examples/st.conf_changes`. You can read in the entire file to `/kernel/drv/st.conf` or you can merge the necessary changes. For more information on updating the `st.conf` file, see the *ASM, ASM-QFS, and ASM/QFS-Standalone Installation and Configuration Guide*.

3. Power-off the host system and peripherals using the manufacturer's suggested procedure.
4. Replace the tape drives with the new drives.
5. Power-on the peripherals and host system using the suggested power-on sequence.
6. Ensure that the host system identifies the new drives.

Enter the following command:

```
> probe-scsi-all
```

The automated library and the new drives must be displayed prior to proceeding. If these devices are not displayed, a connection problem probably exists and needs to be corrected. After this command returns the expected information, you can proceed to the next step.

7. Boot the system.

   Enter the following command to boot with the new configuration:

```
> boot -rv
```

8. Modify the `/etc/opt/SUNWsamfs/mcf` file to reflect the new configuration. (Optional)

   Perform this step if the target numbers of the drives or automated library changed, or if the ordering or number of the drives in the automated library changed. This is similar to an initial installation as described in the *ASM, ASM-QFS, and ASM/QFS-Standalone Installation and Configuration Guide*.

9. Create new `/dev/samst` entries for the new equipment. (Optional)

To create these entries, enter the following command:

```
# samdev
```

10. Start your ASM or ASM-QFS system.
11. Mount the file systems.

You can now continue to use the existing ASM or ASM-QFS tapes.

Advanced Topics

11

This chapter discusses advanced topics beyond the scope of basic system administration and usage.

The following topics are presented.

- Device logging
- Removable media files
- Volume overflow
- Segmented files
- System Error Facility (SEF) reporting

Device Logging

The device-logging facility provides device-specific error information that you can use to analyze certain types of device problems. It can help to determine a failing sequence of events for an automated library, tape drive, or optical drive. Note that the device-logging facility does not collect soft media errors (such as recoverable read errors).

Device-logging messages are written to individual log files. There is a log file for each automated library, each tape and optical drive device, and one for the historian. The log files are located in `/var/opt/SUNWsamfs/devlog`. The name of each individual log file is the same name as the Equipment Ordinal.

Example. Assume an ASM file system and a single Hewlett Packard optical library with two optical drives.

Figure 44. shows the `mcf` file.

Figure 44. Example `mcf` File

```
/dev/samst/c1t5u0  40  hp  hp40  -  etc/opt/SUNWsamfs/
hp40_cat
/dev/samst/c1t4u0  41  mo  hp40  -
/dev/samst/c1t6u0  42  mo  hp40  -
```

The `/var/opt/SUNWsamfs/devlog` file is as follows:

```
# pwd
/var/opt/SUNWsamfs/devlog
# ls
40      41      42      43
#
```

Device 43 is the historian.

When to Use the Device Log

The device log can easily generate many log messages, especially when all logging options for all devices are turned on and there is a great deal of device activity. Initially, the device log settings are set to the following default values:

```
err, retry, syserr
```

If you suspect there is a problem with one of the devices configured within an ASM or ASM-QFS environment, it is appropriate to enable additional logging events for that device. Also, it is appropriate to enable device logging if you are advised to do so by your service provider. In these situations, set the event to `detail`. In extreme cases, you might be advised by your service provider to set the event to `all` for a device. This adds additional log information. However, in general, it is probably not useful or practical to run the system with excessive logging.

The device log information is collected automatically when the `info.sh`(1M) command is issued. This allows the file system service to review any possible device error information as part of problem analysis activity.

Enabling the Device Log

There are two methods you can use to enable the device log.

For both method 1 and method 2:

- *eq* is the Equipment Ordinal of the device from the `mcf` file or the keyword `all` for all equipment.
- The device-logging events are listed in the `samset`(1M) man page. They are also listed in “Enabling the Device Log” on page 270. Note that the device log messages are available only in English text. *event* is one or more event types from the following list:
  - `all`
  - `date`

- default
- detail
- err
- event
- label
- mig
- module
- msg
- none
- retry
- stage
- stage_ck
- syserr
- time

Method 1

Method 1 uses the samset(1M) command in the following format:

```
samset devlog eq event
```

For more information on the samset(1M) command, see the samset(1M) man page.

Method 2

Method 2 requires a directive to the /etc/opt/SUNWsamfs/defaults.conf file. Edit the defaults.conf file, and add the following directive:

```
devlog eq event
```

When an ASM or ASM-QFS file system starts up, it automatically sets the event type for each available device to default. You can also use the samset(1M) command to determine the present settings for each device log.

Removable Media Files

You can use the `request`(1) command to manually create, write, and read files that do not use the disk cache for buffering the data. Files created in this manner are called *removable media files*.

Removable media files look like typical ASM or ASM-QFS files in that they have permissions, a user name, a group name, and size characteristics. However, the data does not reside in the disk cache. Thus, files larger than the disk cache can be created and written to media. An inode entry is created in the `.inodes` file for the file specified in the `request`(1) command. The user does not need to know where the file begins on the removable media. (It is the same for a file with data in the disk cache.) The ASM and ASM-QFS file systems read that information from the inode entry. Multiple removable media files can reside on the same cartridge.

Removable media files must be read and written sequentially. The media type and at least one VSN for the media must be specified. Multiple volumes (up to 256) can be specified to handle volume overflow (see “Volume Overflow” on page 272"). The ASM or ASM-QFS file system automatically mounts the requested volume if the volume resides in an automated library defined in the `mcf` file.

The volumes used for the `request`(1) command should not be the same volumes that are used in an ASM or ASM-QFS environment for automated archiving. Archiving appends the next file to be archived to the end of the current data and moves the EOF label beyond the data each time.

The presence of a removable media file on a volume prevents that volume from being recycled. The recycler expects that only archived files reside on the particular volume that is assigned for archiving. In addition, the removable media files are never archived.

Removable media files are not supported over NFS.

If the `-N` option is specified on the `request`(1) command or on the `sam_request`(3) library routine, the file being read can be one that is foreign to the ASM or ASM-QFS environment. This option can be used when reading tapes that are unlabeled, barcoded, and write-protected. Up to 256 volumes can be specified for volume overflow files.

For examples that describe how to create removable media files, see the `request`(1) man page.

Volume Overflow

Volume overflow allows the system to span a single file over multiple volumes. It is useful for very large files that exceed the capacity of their chosen media. Note that when you use the volume overflow feature, it is difficult to retrieve volume overflow data if you need to retrieve the file because of a disaster. For more information, see the `request`(1) man page.

Volume overflow is enabled when you use the `ovflmin` directive in the `archiver.cmd` file. When a file size exceeds `ovflmin`, the archiver writes another portion of this file to another available volume of the same type, if necessary. The portion of the file written to each volume is called a *section*. For instructions on setting the `ovflmin` directive for volume overflow, see the information on controlling volume overflow in “Archiving” on page 77.

Volume overflow removable media files can be created directly by using the `request`(1) command. Note that using the `request`(1) command bypasses the typical functions of the archiver. When overflowing the file to separate volumes, you must separate VSNs with a slash. It is possible to list the VSNs in a file using the `-l` option to the `request`(1) command. For the complete syntax, see the `request`(1) man page.

Example 1. The following is an example `request`(1) command that creates a removable media file on Ampex D2 tapes using three volumes:

```
# request -m d2 -v TAPE01/TAPE02/TAPE03 large.file
```

Example 2. The `sls`(1) command lists the archive copy showing each section of the file on each VSN. This example shows the archiver log file and the `sls -D` command output for a large file named `file50` that spans multiple volumes.

The archive log file illustrated here shows that `file50` spans three volumes with VSNs of `DLT000`, `DLT001`, and `DLT005`. The position on volume and size of each section is indicated in the seventh and tenth fields respectively, and matches the sls `-D` output also shown. For a complete description of the archiver log entry, see the `archiver`(1M) man page.

The archive log file entry for `file50` is as follows:

```
A  97/01/13  16:03:29  lt  DLT000  big.1  7eed4.1  samfs1
13.7  477609472  00 big/file50  0  0

A  97/01/13  16:03:29  lt  DLT001  big.1  7fb80.0  samfs1
13.7  516407296  01 big/file50  0  1

A  97/01/13  16:03:29  lt  DLT005  big.1  7eb05.0  samfs1
13.7  505983404  02 big/file50  0  2
```

The `sls -D` output is as follows:

```
# sls -D file50
file50:
  mode: -rw-rw----  links:   1  owner: gmm    group: sam
  length: 1500000172  admin id:      7  inode: 1407.5
  offline;   archdone;  stage -n
  copy1: ---- Jan 13 15:55 1e4b1.1    lt DLT001
    section 0:  477609472     7eed4.1    DLT000
    section 1:  516407296     7fb80.0    DLT001
    section 2:  505983404     7eb05.0    DLT005
  access:      Jan 13 17:08  modification: Jan 10 18:03
  changed:     Jan 10 18:12  attributes:   Jan 13 16:34
  creation:    Jan 10 18:03  residence:    Jan 13 17:08
```

Up to 256 volumes can be specified for volume overflow files.

Note that volume overflow files do not generate checksums. For more information on using checksums, see the `ssum`(1) man page.

If you are using volume overflow, and a file you are retrieving spans multiple volumes, see the examples in the *ASM, ASM-QFS, and ASM/QFS-Standalone Disaster Recovery Guide* for information on how to retrieve such a file.

Segmented Files

The ASM and ASM-QFS environments support segmented files. Segmenting files improves tape storage retrieval speed, access, and manageability for very large files. A segmented file can be larger than the physical disk cache. With a segmented file, it is possible for only part of a file to reside on the disk cache at any one time.

The `segment`(1) command allows you to specify the segment size. You cannot set a segment size that is larger than the current file size.

Segmented files support tape striping. After a file is segmented, it can be striped simultaneously over multiple tape devices, which significantly reduces the time needed to store the file segments. Data access is accelerated by allowing users to retrieve only the desired file segments rather than the entire file.

Segmentation can enhance archiving efficiency because only changed portions of a file are rearchived. Segments of a file can be archived in parallel, and segmented files can be staged in parallel. This increases performance when archiving and retrieving.

Segmentation can be enabled on a file, directory, or entire file system. Segmented files support all other ASMand ASM-QFS capabilities.

The following sections describe how segmented files differ from nonsegmented files. For more information on segmented files, see the `segment`(1) or the `sam_segment`(3) man pages.

Archiving

For a segmented file, the archivable unit is the segment itself, not the file. All archiving properties and priorities apply to the individual segments, and not to the file.

The unit archived is the segment. The segment can be striped by specifying the `-drives` parameter for the archive set in the `archiver.cmd` file.

For example, assume that there is a 100-megabyte segmented file in the file system, and its segment size is 10 megabytes. If the `archiver.cmd` file defines an archive set with a `-drives` 2 directive, this file is archived to 2 drives in parallel. Segments 1, 3, 5, 7, and 9 are archived using the first drive, and segments 2, 4, 6, 8, and 10 are archived using the second drive.

Only segments that have been modified are archived—not the entire file. Up to four archive copies can be made for each segment. Volume overflow is supported for segments.

Note: The index of a segmented file contains no user data. It is considered metadata. It is assigned to the file system archive set.

Disaster Recovery

For information on recovering a segmented file in the event of a disaster, see the *ASM, ASM-QFS, and ASM/QFS-Standalone Disaster Recovery Guide.*

■ System Error Facility Reporting

The system error facility (SEF) reporting system captures log sense data from tape devices in an automated library, writes it to a log file, and translates that data into human-readable form. It consists of the following:

- A log file containing data from tape device log sense pages.
- A command, `sefreport`(1M), for writing the log file to `stdout` in a human-readable format. This log file can be used as input to a user-supplied analysis script.

The `sefreport`(1M) command reads the content of a ASM or ASM-QFS SEF log file. The log file contains data gathered from the log sense pages of the peripheral tape devices used in a ASM and ASM-QFS environment. The log sense pages differ from vendor to vendor. For the meanings of the parameter codes, control bits, and parameter values, see the vendor documentation for each specific device.

SEF reporting is not supported for standalone tape drives.

To Enable SEF Reporting

1. Log into the system as `root`.

   You need to become superuser to complete this procedure.

2. Use the `mkdir`(1) command to create the SEF directory.

   For example:

```
# mkdir /var/opt/SUNWsamfs/sef
```

3. Use the `touch`(1) command to enable SEF reporting.

   You can enable SEF reporting any time after installation by creating the `sefdata` log file. Initially, the SEF log file must be empty. You can use the following command to create the file:

```
# touch /var/opt/SUNWsamfs/sef/sefdata
```

The preceding example command shows the SEF log file being created in `/var/opt/SUNWsamfs/sef/sefdata`. This is the default location.

SEF data is appended to the log file as it is generated.

You can configure SEF reporting to log and read log sense data from an alternate location. For more information on reading log sense data from an alternate location, see the `sefreport`(1M) man page.

SEF Report Output

Before you use the `sefreport`(1M) command, ensure that `/opt/SUNWsamfs/sbin` is in your command path. The SEF report output consists of header lines and log sense data.

Following the header lines, the log sense data for each page in the record is printed. For each log sense page, a line identifying the page code is printed, followed by a line of column headings. The data is then printed, three columns per line, with the following headings: `param code`, `control`, and `param value`. All data is generated in hexadecimal notation.

For example, the following SEF command reads the SEF log file from the default location, writes the device number and path name for each device, and generates output:

```
# sefreport -d /var/opt/SUNWsamfs/sef/sefdata >
sef.output
```

Figure 45. shows the content of `sef.output` file.

Figure 45. `sef.output` **Contents**

```
Record no. 1
Mon Mar 26 11:17:48 2001  STK      9840             1.25 VSN
002981
   Eq no. 32   Dev name /dev/rmt/1cbn

   PAGE CODE 2
   param code  control   param value
       00h       74h     0x0
       01h       74h     0x0
       02h       74h     0x0
       03h       74h     0x0
       04h       74h     0x0
       05h       74h     0x40050
       06h       74h     0x0

   PAGE CODE 3
   param code  control   param value
       00h       74h     0x0
       01h       74h     0x0
       02h       74h     0x0
       03h       74h     0x0
       04h       74h     0x0
       05h       74h     0x140
       06h       74h     0x0

   PAGE CODE 6
   param code  control   param value
       00h       74h     0x0

Record no. 2
Mon Mar 26 11:30:06 2001  STK      9840             1.25 VSN
002999
   Eq no. 31   Dev name /dev/rmt/0cbn

   PAGE CODE 2
   param code  control   param value
       00h       74h     0x0
       01h       74h     0x0
       02h       74h     0x0
       03h       74h     0x0
       04h       74h     0x0
       05h       74h     0x1400a0
       06h       74h     0x0
```

Figure 45. `sef.output` **Contents (Continued)**

```
   PAGE CODE 3
   param code  control   param value
       00h       74h     0x0
       01h       74h     0x0
       02h       74h     0x0

       03h       74h     0x0
       04h       74h     0x0
       05h       74h     0x190
       06h       74h     0x0

   PAGE CODE 6
   param code  control   param value
       00h       74h     0x0

Record no. 3
Mon Mar 26 11:30:23 2001  STK      9840             1.25 VSN
002981
   Eq no. 32   Dev name /dev/rmt/1cbn

   PAGE CODE 2
   param code  control   param value
       00h       74h     0x0
       01h       74h     0x0
       02h       74h     0x0
       03h       74h     0x0
       04h       74h     0x0
       05h       74h     0x18400f0
       06h       74h     0x0


   PAGE CODE 3
   param code  control   param value
       00h       74h     0x0
       01h       74h     0x0
       02h       74h     0x0
       03h       74h     0x0
       04h       74h     0x0
       05h       74h     0x1e0
       06h       74h     0x0

   PAGE CODE 6
   param code  control   param value
       00h       74h     0x0
.
.
.
```

Note: The preceding output has been truncated for inclusion in this manual.

For more information on the SEF log file, including its content and format, see the `sefdata`(4) man page. For more information on optional SEF report formats, see the `sefreport`(1M) man page.

The sefreport(1M) Command

The following are the two most commonly used options on the `sefreport`(1M) command:

- The `-d` option. The `-d` option generates additional device information. It writes an additional header line that contains the Equipment Ordinal and path name to the device for each record. This makes it easier to search for and to locate SEF records that pertain to a specific device.
- The `-v` option. The `-v` option generates information in verbose mode. It appends information regarding the Equipment Ordinal, page code, and VSN to each line of a record. This makes it possible to select only those lines that pertain to a specific device or a specific volume.

Managing the SEF Log File

The SEF log file can be managed just as any other ASM or ASM-QFS log file is managed. A `cron`(1) job can be run periodically to save the current log file to another location, to delete old SEF files, to create new (empty) SEF files, or to perform other tasks.

You can also use the `log_rotate.sh`(1M) utility to rotate this log file.

For more information on tools for managing the SEF log file, see the `cron`(1) or `log_rotate.sh`(1M) man pages.

Glossary

A

addressable storage

The storage space encompassing online, nearline, offsite, and offline storage that is user-referenced through an ASM or ASM-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.

archive storage

Copies of file data that have been created on archive media.

archiver

The archive program that automatically controls the copying of files to removable cartridges.

audit (full)

The process of loading cartridges to verify their VSNs. For magneto-optical cartridges, the capacity and space information is determined and entered into the automated library's catalog.

automated library

A robotically controlled device designed to automatically load and unload removable media cartridges without operator intervention. An automated library contains one or more drives and a transport mechanism that moves cartridges to and from the storage slots and the drives.

B

backup storage

A snapshot of a collection of files for the purpose of preventing inadvertent loss. A backup includes both the file's attributes and associated data.

block allocation map

A bitmap representing each available block of storage on a disk and indicating whether the block is in use or free.

block size

See DAU.

C

cartridge

A physical entity that contains media for recording data. A tape or optical disk. Sometimes referred to as *a piece of media*, *a volume*, or *the medium*.

catalog

A record of the VSNs in an automated library. There is one catalog for each automated library, and at a site, there is one historian for all automated libraries.

client-server

The model of interaction in a distributed system in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called the client. The program satisfying the response is called the server.

connection

The path between two protocol modules that provides reliable stream delivery service. A TCP connection extends from a TCP module on one machine to a TCP module on the other.

D

data device

For a QFS, ASM, or ASM-QFS 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.

The ASM and ASM-QFS file systems support both a small and a large DAU. The small DAU is 4 kilobytes (2^{14} or 4096 bytes). The large DAU is 16, 32, or 64 kilobytes. The available DAU size pairs are 4/16, 4/32, and 4/64.

In addition, the QFS and ASM-QFS file systems support a fully adjustable DAU, sized from 16 kilobytes through 65,528 kilobytes. The DAU you specify must be a multiple of 8 kilobytes.

device logging

A configurable feature that provides device-specific error information used to analyze device problems.

device scanner

Software within the ASM or ASM-QFS file system that periodically monitors the presence of all manually mounted removable devices and that detects the presence of mounted cartridges that can be requested by a user or other process.

direct access

A file attribute (stage never) designating that a nearline file can be accessed directly from the archive media and need not be retrieved to disk cache.

direct-attached library

An automated library connected directly to a server using a SCSI interface. A SCSI attached library is controlled directly by the ASM or ASM-QFS software by using the SCSI standard for automated libraries.

direct I/O

An attribute used for large block-aligned sequential I/O. The `setfa`(1) command's `-D` option is the direct I/O option. It sets the direct I/O attribute for a file or directory. If applied to a directory, the direct I/O attribute is inherited.

directory

A file data structure that points to other files and directories within the file system.

disk allocation unit

See DAU.

disk buffer When using ASM-Remote software, the disk buffer is a buffer on the server system that is used when archiving data from the client to the server.

disk cache The disk-resident portion of the ASM and ASM-QFS file system software. It is used to create and manage data files between online disk cache and archive media. Individual disk partitions or an entire disk can be used as disk cache.

disk space thresholds An administrator-defined amount of disk space that is available to a user. This defines the range of desirable disk cache utilization. The high threshold indicates the maximum level of disk cache utilization. The low threshold indicates the minimum level of disk cache utilization. The releaser controls disk cache utilization based on these predefined disk space thresholds.

disk striping The process of recording a file across several disks, thereby improving access performance and increasing overall storage capacity. Also see entries for striping.

drive A mechanism for transferring data to and from a removable media volume.

E

Ethernet

A local-area, packet-switched network technology. Originally designed for coaxial cable, it is now found running over shielded, twisted-pair cable. Ethernet is a 10- or 100-megabytes-per-second LAN.

extent array

The array within a file's inode that defines where each data block assigned to the file is located on the disk.

F

family device set

See family set.

family set

A storage device that is represented by a group of independent physical devices, such as a collection of disks or the drives within an automated library. Also see disk cache family set.

FDDI

Fiber distributed data interface. A 100-megabytes-per-second fiber-optic LAN.

fibre channel

The ANSI standard that specifies high-speed serial communication between devices. Fibre channel is used as one of the bus architectures in SCSI-3.

fibre-distributed data interface

See FDDI.

file system

A hierarchical collection of files and directories.

file system specific directives

Archiver and releaser directives that follow global directives, are specific to a particular file system, and begin with `fs =`. File system specific directives apply until the next `fs =` directive line or until the end of file is encountered. If multiple directives affect a file system, the file system-specific directives override the global directives.

FTP

File Transfer Protocol. An internet protocol for transferring files between two hosts over a TCP/IP network.

G

global directives

Archiver and releaser directives that apply to all file systems and that appear before the first `fs =` line.

grace period

For disk quotas, this is the amount of time that can elapse during which a user is allowed to create files and/or allocate storage after a user reaches their soft limit.

H

hard limit

For disk quotas, a maximum limit on file system resources (blocks and inodes) that users cannot exceed.

I

indirect block

A disk block that contains a list of storage blocks. The QFS, ASM, and ASM-QFS file systems have up to three levels of indirect blocks. A first-level indirect block contains a list of blocks used for data storage. A second-level indirect block contains a list of first-level indirect blocks. A third-level indirect block contains a list of second-level indirect blocks.

inode

Index node. A data structure used by the file system to describe a file. An inode describes all the attributes associated with a file other than the name. The attributes include ownership, access, permission, size, and the file location on the disk system.

inode file

A special file (`.inodes`) on the file system that contains the inode structures for all files resident in the file system. All QFS, ASM, and ASM-QFS inodes are 512 bytes long. The inode file is a metadata file, which is separated from file data in the QFS and ASM-QFS file systems.

K

kernel

The central controlling program that provides basic system facilities. The UNIX kernel creates and manages processes, provides functions to access the file system, provides general security, and supplies communication facilities.

L

LAN

Local area network.

lease

In a QFS shared file system, a lease grants a client host permission to perform an operation on a file for as long as the lease is valid. The metadata server issues leases to each client host. The leases are renewed as necessary to permit continued file operations.

library

See automated library.

library catalog

See catalog.

LUN

Logical unit number.

M

mcf

Master configuration file. The file that is read at initialization time that defines the relationships between the devices (the topology) within a QFS, ASM, and ASM-QFS environment.

media

Tape or optical disk cartridges.

media recycling

The process of recycling or reusing archive media with low use (that is, archive media with few active files).

metadata

Data about data. Metadata is the index information needed to locate the exact data position of a file on a disk. It consists of information about files, directories, access control lists, symbolic links, removable media, segmented files, and the indexes of segmented files. Metadata must be protected because if data is lost, the metadata that locates the data must be restored before the lost data can be retrieved.

metadata device

A separate device (for example, a solid-state disk or mirrored device) upon which QFS and ASM-QFS file system metadata is stored. Separating file data from metadata can increase performance. In the `mcf` file, a metadata device is declared as an `mm` device within an `ma` file system.

mirror writing

The process of maintaining two copies of a file on disjointed sets of disks to prevent loss from a single disk failure.

mount point

The directory on which a file system is mounted.

multireader file system

The QFS multireader file system is a single-writer, multireader capability that enables you to specify a file system that can be mounted on multiple hosts. Multiple hosts can read the file system, but only one host can write to the file system. Multiple readers are specified with the `-o reader` option on the `mount`(1M) command. The single-writer host is specified with the `-o writer` option on the `mount`(1M) command. For more information on the `mount`(1M) command, see the `mount_samfs`(1M) man page.

N

name space

The metadata portion of a collection of files that identifies the file, its attributes, and its storage locations.

nearline storage

Removable media storage that requires robotic mounting before it can be accessed. Nearline storage is usually less expensive than online storage, but it incurs a somewhat longer access time.

network-attached automated library

A library, such as those from StorageTek, ADIC/Grau, IBM, or Sony, that is controlled using a software package supplied by the vendor. The ASM and ASM-QFS file systems interface with the vendor software using a ASM or ASM-QFS media changer daemon designed specifically for the automated library.

NFS

Network file system. An ASM distributed file system that provides transparent access to remote file systems on heterogeneous networks.

NIS

The SunOS 4.0 (minimum) Network Information Service. A distributed network database containing key information about the systems and the users on the network. The NIS database is stored on the master server and all the slave servers.

O

offline storage

Storage that requires operator intervention for loading.

offsite storage

Storage that is remote from the server and is used for disaster recovery.

online storage

Storage that is immediately available (for example, disk cache storage).

P

partition

A portion of a device or a side of a magneto-optical cartridge.

preallocation

The process of reserving a contiguous amount of space on the disk cache for writing a file. This ensures that the space is contiguous. Preallocation can be performed only on zero-sized files. That is, the `setfa -l` command can be specified only for a file that is size zero. For more information, see the `setfa`(1) man page.

prioritizing preview requests

Assigning priority to archive and stage requests that cannot be immediately satisfied.

pseudo device

A software subsystem or driver with no associated hardware.

Q

quota

The amount of system resources that a user is allowed to consume. Quotas are not supported for removable media or disk archive resources.

R

RAID

Redundant array of inexpensive/ 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 ASM and ASM-QFS utility that reclaims space on cartridges that is occupied by expired archive copies.

release priority

A method of calculating the release priority of a file within a file system by multiplying various weights by the corresponding file properties and then summing the results.

releaser

A ASM and ASM-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 to high and low thresholds.

remote procedure calls

See RPC.

removable media file

A special type of user file that can be accessed directly from where it resides on a removable media cartridge, such as magnetic tape or optical disk cartridge. also used for writing archive and stage file data.

robot The portion of an automated library that moves cartridges between storage slots and drives. Also called a transport.

round robin

A data access method in which entire files are written to logical disks in a sequential fashion. When a single file is written to disk, the entire file is written to the first logical disk. The second file is written to the next logical disk, and so on. The size of each file determines the size of the I/O.

By default, QFS, ASM, and ASM-QFS file systems implement striped data access unless striped groups are present. Files are round-robined if round robin access is specified. If the file system contains mismatched striped groups, striping is not supported and round robin is forced.

Also see glossary entries for disk striping and striping.

RPC

Remote procedure calls. The underlying data exchange mechanism used by NFS to implement custom network data servers.

S

samfsdump

A program that creates a control structure dump and copies all the control structure information for a given group of files. It is analogous to the UNIX `tar`(1) utility, but it does not generally copy file data.

samfsrestore

A program that restores inode and directory information from a control structure dump.

SCSI

Small Computer System Interface. An electrical communication specification commonly used for peripheral devices such as disk and tape drives and automated libraries.

small computer system interface

See SCSI.

soft limit

For disk quotas, a threshold limit on file system resources (blocks and inodes) that you can temporarily exceed. Exceeding the soft limit starts a timer. When you exceed the soft limit for the specified time (default is one week), no further system resources can be allocated until you reduce file system use to a level below the soft limit.

staging

The process of copying a nearline or offline file from archive storage back to online storage.

storage family set

A set of disks that are collectively represented by a single disk family device.

storage slots

Locations inside an automated library in which cartridges are stored when not being used in a drive. If the library is direct-attached, the contents of the storage slots are kept in the automated library's catalog.

stripe size

The number of disk allocation units (DAUs) to allocate before moving to the next device of a stripe. If `stripe=0`, the file system uses round-robin access, not striped access.

striped group

A collection of devices within a QFS or ASM-QFS file system and defined in the `mcf` file as one (usually two) or more `g`*XXX* devices. Striped groups are treated as one logical device and are always striped with a size equal to the disk allocation unit (DAU). You can specify up to 128 striped groups within a file system, but you can specify no more than 252 total devices.

striping

A data access method in which files are simultaneously written to logical disks in an interlaced fashion. All QFS, ASM, and ASM-QFS file systems enable you to declare either striped or round robin access for each individual file system. The QFS and ASM-QFS file systems enable you to declare striped groups within each file system. Also see the glossary entry for round robin.

ASM

The Storage and Archive Manager File System. The ASM software controls the access to all files stored and all devices configured in the master configuration file (`mcf`).

ASM-QFS

The ASM-QFS software combines the Storage and Archive Manager with the QFS file system. ASM-QFS offers a high-speed, standard UNIX file system interface to users and administrators in conjunction with the storage and archive management utilities. It uses many of the commands available in the ASM command set as well as standard UNIX file system commands.

ASM-Remote client

A ASM-Remote client is a ASM or ASM-QFS system that establishes a ASM-Remote client daemon that contains a number of pseudodevices. It might or might not have its own library devices. The client depends on a ASM-Remote server for archive media for one or more archive copies.

ASM-Remote server

The ASM-Remote server is both a full-capacity ASM or ASM-QFS storage management server and a ASM-Remote server daemon that defines libraries to be shared among ASM-Remote clients.

superblock

A data structure in the file system that defines the basic parameters of the file system. It is written to all partitions in the storage family set and identifies the partition's membership in the set.

T

tar

Tape archive. A standard file/data recording format used by the ASM and ASM-QFS software for archive images.

TCP/IP

Transmission Control Protocol/Internet Protocol. The internet protocols responsible for host-to-host addressing and routing, packet delivery (IP), and reliable delivery of data between application points (TCP).

thresholds

A mechanism for defining the desirable available storage window for online storage. Thresholds set the storage goals for the releaser. Also see disk space thresholds.

timer

Quota software that keeps track of the time elapsed between a user reaching a soft limit and a hard limit being imposed on the user.

V

volume

A named area on a cartridge for sharing data. A cartridge has one or more volumes. Double-sided cartridges have two volumes, one on each side.

volume overflow

A capability that enables the system to span a single file over multiple volumes. Volume overflow is useful for sites using very large files that exceed the capacity of their individual cartridges.

VSN

Volume serial name. If you are archiving to removable media cartridges, the VSN is a logical identifier for magnetic tape and optical disk that is written in the volume label. If you are archiving to disk cache, this is the unique name for the disk archive set.

W

WORM

Write once read many. A storage classification for media that can be written only once but read many times.

Index

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

V

W

Reader’s Comment Form

Contact Us

Submit your questions, comments, and suggestions to StorageTek’s Global Learning Solutions. We appreciate your correspondence and are committed to responding to you.

Publication Information

Publication Name:

Publication Part Number:

Questions and Comments:

Note: Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form. If you would like a reply, please supply your name and address on the reverse side of this form.

Thank you for your cooperation. No postage stamp is required if mailed in the U.S.A.

TO COMPLY WITH POSTAL REGULMATIONS, FOLD EXACTLY ON DOTTED LINES AND TAPE (DO NOT

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 2 LOUISVILLE, CO U.S.A.

POSTAGE WILL BE PAID BY ADDRESSEE

GLOBAL LEARNING SOLUTIONS MS 3256
STORAGE TECHNOLOGY CORPORATION
ONE STORAGETEK DRIVE
LOUISVILLE CO 80028-9989
USA

FOLD HERE AND TAPE DO NOT STAPLE FOLD HERE AND TAPE

If you would like a reply, please print:

Your Name:____

Company Name:_____________________________ Department:_____________

Street Address:____

City:____

State:____________________________________ Zip Code:______________

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

NEED MORE INFORMATION?

www.storagetek.com

STORAGETEK®

ABOUT STORAGETEK®

StorageTek® (NYSE:STK), a $2 billion worldwide company with headquarters in Louisville, Colo., delivers a broad range of storage solutions for digitized data. StorageTek solutions are easy to manage and allow universal access to data across servers, media types and storage networks. StorageTek is the innovator and global leader in virtual storage solutions for tape automation, disk storage systems and storage networking. Because of StorageTek, customers can manage and leverage their digital assets as their businesses grow, and can maximize IT productivity to ensure enterprise-class business continuity.

WORLD HEADQUARTERS
Storage Technology Corporation
One StorageTek Drive
Louisville, Colorado 80028 USA
Phone: **1.800.525.0369**