StorageTek

Application Storage Manager™ (ASM) for Unix

Administrator's Guide

Version 3.5.0

Part Number 313498401

Proprietary Information Statement

Limitations on Warranties and Liability

Restricted Rights

Application Storage Manager™ (ASM) for Unix – Administrator's Guide, Version 3.5.0, June 7, 2002, Part 313498401

This edition applies to the Application Storage Manager™ (ASM) for Unix product and to all modifications of that product until otherwise indicated in new editions or revisions pages. If there are changes in the product or improvements in the information about the product, this document will be revised and reissued.

Comments concerning the contents of the manual should be directed to:

ASM Product Management
Storage Technology Corporation
One StorageTek Drive, MS 2129
Louisville, CO 80028-2129

New Features

The *ASM Administrator's Guide*, revision 3.5.0, supports the ASM and ASM-QFS 3.5.0 releases running on Solaris 2.6, 2.7, and 2.8 platforms. The StorageTek ASM 3.5.0 releases support the following new features in the ASM and ASM-QFS environments:

Solaris 2.8 operating system. StorageTek products can now be installed on servers running the Solaris 2.8 operating system.

ASM-QFS Share. The ASM-QFS file system can be used in conjunction with fiber-attached devices in a Storage Area Network (SAN). When the ASM-QFS Share license key from StorageTek is enabled, the ASM-QFS file system enables high-speed access to data using software such as Tivoli SANergy File Sharing.

ACSLS 5.4. The ASM and ASM-QFS environments now support the StorageTek ACSLS 5.4 release for network-attached StorageTek automated libraries.

New devices. The ASM and ASM-QFS environments now support the following devices:

Device	Comments
Exabyte Mammoth-2 drive	Equipment type xm in your StorageTek mcf file.
Fujitsu M8100 drive	Equipment type fd in your StorageTek mcf file.
Sony DTF-2 drive	Equipment type so in your StorageTek mcf file.
Sony network-attached automated library and drives	Sites with Sony network-attached automated libraries use the Sony DZC-8000S interface. Such automated libraries are equipment type pe in your StorageTek mcf file. The drives are equipment type so in your mcf file. Sites with automated libraries attached through this interface need to add the StorageTek sony (samsony) package at installation time.
StorageTek 9940 drive	Equipment type sf in your StorageTek mcf file.

Documentation restructuring. The StorageTek manual set has been restructured in order to make the documentation more modular. Certain parts of the original *ASM System Administrator's Guide* have been moved into new manuals or man pages.

The following table indicates the topic that was affected by this restructuring, where it used to reside in the StorageTek documentation set prior to the 3.5.0 release, and where it resides in the documentation set as of the 3.5.0 release:

Topic	Pre-3.5.0 Location	3.5.0 Location
ASM-QFS installation and configuration.	ASM-QFS Administrator's Guide	ASM Installation and Configuration Guide
ASM installation and configuration.	ASM System Administrator's Guide, chapter 2.	ASM Installation and Configuration Guide
ASM-QFS file system overview, reference, and operations information.	ASM-QFS Administrator's Guide	ASM File System Administrator's Guide
ASM file system overview, reference, and operations information	*ASM System Administrator's Guide*, chapter 5 and part of chapter 14.	ASM Administrator's Guide
Application Programmer Interface (API) overview	ASM System Administrator's Guide, chapter 12.	intro_libsam(3) man page.
Storage and archive management operational, reference, and disaster recovery information	*ASM System Administrator's Guide*, chapters 3, 4, 6, 7, 8, 9, 10, 11, 13, and 14; appendixes A and B.	ASM Administrator's Guide

Record of Revision

Revision	Description
3.5.0	November 2000. Original printing.
3.5.0	July 2002. Document update.

Table of Contents

Preface

This manual describes the storage and archive management software supported within the ASM and ASM-QFS environments. The ASM and ASM-QFS 3.5.0 releases are supported by StorageTek on the Solaris 2.6, 2.7, and 2.8 operating system platforms.

The *ASM and ASM-QFS Administrator's Guide* is written for system administrators responsible for setting up and maintaining ASM and ASM-QFS software. You, the system administrator, are assumed to be knowledgeable about Solaris operating system procedures, including creating accounts, performing system backups, and other basic Solaris system administrator tasks.

Other StorageTek software products, such as ASM-Remote, can be licensed for use within the ASM or ASM-QFS environment. For more information on this and other StorageTek products, see the Licensing subsection in this preface.

This manual is organized as follows:

Section	Title
Chapter 1	Overview
Chapter 2	Basic Operations
Chapter 3	Managing Vendor-specific Automated Libraries
Chapter 4	The Archiver
Chapter 5	The Stager
Chapter 6	The Releaser
Chapter 7	The Recycler
Chapter 8	Graphical User Interface (GUI) Tools
Chapter 9	The ASM (1M) Operator Utility

Section	Title
Chapter 10	How to Upgrade Your Environment
Chapter 11	Advanced Features
Appendix A	StorageTek Product Support
Appendix B	Control Structures, Disaster Preparation, and Recovery

In addition to the preceding sections, the glossary section defines terms used in the ASM and ASM-QFS documentation.

NOTE

For users familiar with the StorageTek documentation set prior to the StorageTek 3.5.0 releases, please note that several topics that used to be included in the *ASM System Administrator's Guide*, have moved to other manuals.

The storage and archive management information is in this manual; the installation and configuration information has moved to the *QFS, ASM, and ASM-QFS Installation and Configuration Guide*; the file system information has moved to the *StorageTek File System Administrator's Guide*, and the Application Programmer Interface (API) information has moved to the intro_libASM(3) man page.

Conventions

The following conventions and terms are used throughout this manual:

Convention	Meaning
Courier	The fixed-space courier font denotes literal items such as commands, files, path names, system prompts, system output, and messages. For example: /etc/opt/SLSCSAMfs/mcf
Bold courier	The **bold courier** font denotes text you enter at the shell prompt. For example: server# **sls –D**
[ ]	Brackets enclose optional portions of commands or optional arguments to commands.

Convention	Meaning
Italic	Italics indicate either a variable or a term being defined. For a variable, you must replace the variable with a real name or value. For example: server# **mount** *mnt_pt*
|	The pipe symbol indicates that one of two or more optional arguments must be specified.

Certain terms are used throughout this manual. Many terms can be found in the glossary, but some of the most commonly used ones are as follows:

Term	Meaning
Archiving	Automatically copying online, magnetic disk cache files to archive media.
Automated library	An automated device for storing tape and optical cartridges.
Cartridge	A tape or magneto optical cartridge.
Partition	A side of a magneto optical disk or a partition on an Ampex tape.
Staging	Automatically copying files located on archive media back to online disk.
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.

Other StorageTek Publications

In addition to this manual, the following StorageTek publications might be useful to you:

- ASM for Unix Migration Toolkit Guide
- ASM-Remote Administrator's Guide
- ASM Man Page Reference Manual
- ASM File System Administrator's Guide
- ASM, and ASM-QFS Installation and Configuration Guide

- Peripherals and Third-Party Software Support:

  http://www.support.storagetek.com/globalnavigation/support/generalpublic/default.htm

To order additional manuals, please send us a request using one of the methods described in the “Reader Comments” subsection.

Other File System Publications

In addition to publications from StorageTek, the following publications on UNIX file systems might interest you:

- Filesystem Hierarchy Standard (FHS) web pages at the following URL: http://www.pathname.com/fhs/2.0/fhs-toc.html

- StorageTek online documentation web pages at the following URL: http://www.storagetek.com./ Go to the Customer Resource Center (CRC) in the Services and Support section.

Licensing

Licenses for StorageTek products can be obtained from StorageTek. In some cases, the capabilities that these additional licenses can provide are described in the ASM or ASM-QFS documentation because these products can be used in within those environments. For information on obtaining licenses for StorageTek software, contact your sales representative, your Authorized Service Provider (ASP), or StorageTek.

Each of the following StorageTek products are licensed separately:

- ASM for Unix Migration Toolkit

- ASM-QFS Standalone

- ASM

- ASM-QFS

- ASM Remote Client

- ASM Remote Server

- ASM Segment

- ASM-QFS Share

Reader Comments

If you have comments about the technical accuracy, content, or organization of this document, please let us know. We value your comments and will respond to them promptly. You can contact us by one of the following methods:

- Send email to info@StorageTek.com
- Send written comments to the following address:

  ASM Product Management
StorageTek,
One StorageTek Drive
MS 2129
Louisville, CO 80028-2129
USA

Chapter 1 - Overview

The ASM and ASM-QFS environments provide a configurable file system with storage, archive management, and retrieval capabilities. Within the ASM or ASM-QFS environment, the system copies files from online disk cache to less expensive automated storage devices, releases disk space associated with archived files, and restores the files to online disk when they are needed. In addition, the ASM and ASM-QFS systems automatically maintain online free disk space at site-specified usage thresholds.

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

- Capabilities
- Storage devices
- Processes and commands

Capabilities

The ASM environment includes the standard StorageTek file system and the Configuration and Installation software. The ASM-QFS environment includes the QFS file system. Both StorageTek file systems are high-performance UNIX file systems that reside in the server's disk cache. The major difference between the two file systems is that the QFS file system offers more high-performance features. For more information on StorageTek file systems, see the *ASM 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 releaser, which automatically maintains the file system's magnetic disk cache at site-specified percentage usage thresholds by freeing disk blocks occupied by eligible archived files.
- The stager, which restores files to the disk cache. When a user or process requests files that have been released from disk cache, the stager automatically copies the files back to the online disk cache.

- The recycler, which clears archive volumes of expired archive copies and makes volumes available for reuse.

The following subsections 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 StorageTek file system and copies the files to secondary storage. You can configure the archiver to create up to four archive copies on a variety of media. The archiving process is initiated when disk-based files match a set of criteria. For more information on the archiver, see chapter 4, “Archiver Operations.”

Staging

When an offline file is accessed, the file system automatically stages the file back to disk cache. For a sequential read of an offline file, the read operation tracks along directly behind the staging operation, allowing the file to be immediately available to an application before completely staging the entire file.

The ASM and ASM-QFS software process stage request errors automatically. If a stage error is returned, the system attempts to find the next available 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 chapter 5, “Staging.”

Releasing

Releasing is the process of freeing primary (disk) storage that is used by a file. Two threshold values, both expressed as a percent of total disk space, are used to manage online disk cache use. These thresholds are the high water mark and the low water mark. When online disk space exceeds the high water mark factor, the system automatically begins releasing the disk space of archived files. Online disk space occupied by archived files is released until the low water mark percentage is reached. Whether or not a file is selected for release depends on the file’s size, age, and other site-configurable factors. For more information on the releaser, see chapter 6, “The Releaser.”

Recycling

As users modify files, 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 moving non-expired copies to different volumes. If only expired copies exist on a given volume, a site-defined action is taken. For example, the volume can be relabeled for immediate reuse or exported to offsite storage, thus keeping an historical record of file changes. Users are unaware of the recycling process as it relates to their data files. For more information on recycling, see chapter 7, “The Recycler”.

Storage Devices

The ASM and ASM-QFS environments support a wide variety of optical and tape storage devices. For a list of supported storage devices, refer to *Peripherals and Third-Party Software Supported*, which is available from the following url: http://www.support.StorageTek.com/globalnavigation/support/generalpublic/default.htm

The topology of the equipment managed within the ASM or ASM-QFS environment is defined in the master configuration file, /etc/opt/LSCsamfs/mcf. This file specifies the devices, automated 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 differentiate between manually mounted devices and automated library devices.

When possible, the system uses the standard Solaris disk and tape device drivers. For devices not directly supported under Solaris, such as certain automated library and optical disk devices, StorageTek provides special device drivers in the ASM and ASM-QFS software packages.

Processes and Commands

The ASM and ASM-QFS environments consist of a file system, processes, various types of commands (user, administrator, and so on), and tools. All the commands are documented in UNIX man(1) pages.

Processes

sam-initd is the master process. If there is at least one StorageTek file system mounted, /etc/rc3.d/S95samd automatically starts sam-initd. However, you can also start sam-initd from the command line by entering the following command:

server# samd start

Only one copy of sam-initd can be running at one time.

All ASM and ASM-QFS processes are named in the form *process_name*d, which is the process name followed by the lowercase letter d. This convention allows ASM and ASM-QFS processes to be easily identified.

The following processes are also initiated by sam-initd:

Process	Description
sam-archiverd	Automatically archives ASM and ASM-QFS files. This process runs as long as any StorageTek file system is mounted.
sam-robotsd	Starts and monitors media changer control daemons.
sam-scannerd	Monitors all manually mounted devices. The scanner periodically checks each device for inserted cartridges.
sam-releaser	Attempts to release disk space of archived files on StorageTek file systems until a low water mark is reached. The releaser is started automatically when a high water mark is reached on disk cache.
sam-stagealld	Controls the associative staging of ASM and ASM-QFS files.
sam-rpcd	Remote procedure call (RPC) Application Programmer Interface (API) server process.

The samd(1M) command is used to start and stop sam-initd and associated processes. Other daemons and processes that may be running on your system include: sam-notifyd, sam-genericd, sam-logd, and sam-catserverd.

User Commands

By default, StorageTek file system operations are transparent to the end user. Depending on your site practices, however, it may be desirable to make some commands available to users at your site. The following commands can help users fine tune certain operations:

Command	Description	Used By
archive(1)	Sets archive attributes on files.	ASM, ASM-QFS

Command	Description	Used By
exarchive(1)	Manipulates (exchanges) archive copies.	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 and has been extended for use with the ASM and ASM-QFS file systems.	ASM, ASM-QFS
segment(1)	Sets segment file attributes. Segmented file features are enabled through the ASM-Segment license key.	ASM, ASM-QFS
setfa(1)	Sets file attributes.	ASM, ASM-QFS
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, 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, ASM-QFS
ssum(1)	Sets the checksum attributes on files.	ASM, ASM-QFS
stage(1)	Copies offline files to disk and sets stage attributes on files.	ASM, ASM-QFS

General System Administrator Commands

The file system provides system management and maintenance commands for administering the system. The following table summarizes the general system administrator commands:

Command	Description	Used By
samcmd(1M)	Executes a single operator utility command.	ASM, ASM-QFS
samd(1M)	Starts or stops daemons.	ASM, ASM-QFS
samfsinfo(1M)	Displays information about a file system.	QFS, ASM, ASM-QFS
samu(1M)	Invokes the curses-based operator interface. Displays the status of devices and allows the operator to control automated libraries.	ASM, ASM-QFS

File System Commands

The file system commands are used to maintain file system operations. These commands are as follows:

Command	Description	Used By
mount(1M)	Mounts a file system. The man page name for this command is mount_samfs(1M).	ASM, ASM-QFS
qfsdump(1M) qfsrestore(1M)	Creates/restores a dump file of the control structures associated with a QFS file system.	ASM-QFS
samfsck(1M)	Checks and repairs a file system.	ASM, ASM-QFS
samfsdump(1M) samfsrestore(1M)	Creates/restores a dump file of the control structures associated with an ASMor ASM-QFS file system.	ASM, ASM-QFS
samgrowfs(1M)	Expands a file system by adding disk partitions.	ASM, ASM-QFS
sammkfs(1M)	Makes a new file system from disk partitions.	ASM, ASM-QFS

Command	Description	Used By
samncheck(1M)	Returns a full path name given the mount point and inode number.	ASM, ASM-QFS

For more information on the StorageTek file systems, see the *ASM File System Administrator's Guide*.

Automated Library Commands

The automated library commands are used to configure, initialize, and maintain the automated libraries and devices within the ASM and ASM-QFS environments:

Command	Description
auditslot(1M)	Audits a single slot within a specified automated library.
build_cat(1M)	Builds a library catalog file for an automated library.
chmed(1M)	Sets or clears library catalog flags and values.
cleandrive(1M)	Requests that a tape drive be loaded with a cleaning tape.
d2format(1M)	Formats and labels Ampex D2 tapes.
dump_cat(1M)	Displays the content of the binary catalog file.
import(1M) export(1M)	Imports or exports cartridges from an automated library by placing it in the mailbox.
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 entries for automated libraries, optical disks, and tape drives attached to the system.
tplabel(1M)	Labels tapes for use with the ASM and ASM-QFS systems.

Archiver Commands

The following commands control the archiver's actions within the ASM and ASM-QFS environments:

Command	Description
archive_audit(1M)	Generates an audit of all archived files.

Command	Description
archiver(1M)	Evaluates the archiver commands file for completeness and accuracy.
showqueue(1M)	Displays the content of an archiver queue file.

Specialized Maintenance Commands

The following table lists various maintenance commands:

Command	Description
dev_down.sh(4)	Sends email to root when a device is marked down or off.
dmpshm(1M)	Dumps the shared memory segments.
info.sh(1M)	Creates a system information file used to troubleshoot the ASM or ASM-QFS system.
itemize(1M)	Catalogs an optical disk.
load(1M) unload(1M)	Load or unloads a cartridge for a specified device.
rearch(1M) unrearch(1M)	Marks or unmarks archive entries to be rearchived.
recycler(1M)	Reclaims space used by expired archive copies from removable media cartridges.
releaser(1M)	Releases disk space.
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.
set_admin.sh(1M)	Adds or removes permission for an administrator group.
setsyscall(1M)	Changes the system call in the ASM or ASM-QFS ...

Command	Description
	library.
set_state(1M)	Sets the state of an ASM or ASM-QFS device.
star(1M)	Creates tape archives and adds or extracts files. This is the StorageTek version of the tar(1) command. It is based on the 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 (API)

The API allows file system requests to be made from within a user application. The requests can be made locally or remotely to the machine upon which the file system is running. The API consists of the libsam and libsamrpc libraries. These libraries contain StorageTek 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.

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 StorageTek GUI tools can help you perform basic operations. The GUI tools are based on Java technology. These tools are as follows:

GUI Tool	Description
devicetool(1M)	Displays and manages devices.
libmgr(1M)	Displays information about and assists in managing automated libraries, devices, and mount requests within the ASM or ASM-QFS environment.

GUI Tool	Description
previewtool(1M)	Displays and manages pending mount requests.
robottool(1M)	Displays and manages automated libraries and tracks exported media.
samtool(1M)	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.

Chapter 2 - Basic Operations

This chapter describes the basic operations for automated libraries and manually loaded devices. The following topics are presented:

- Conventions
- Basic operations
- Automated library operations - systems with a mailbox
- Automated library operations - systems without a mailbox
- Manually loaded drive operations

Some automated libraries have vendor-specific features that cause certain operations to differ from the ones described in this chapter. For example, if you have an automated library that is network attached - not attached through a Small Computer System Interface (SCSI) - some operations may differ. To determine whether your automated library has vendor-specific operating instructions when used in a ASM or ASM-QFS environment, check chapter 3, “Managing Vendor-specific Automated Libraries”, and check the respective man pages for stk(7), grauaci(7), ibm3494(7), fujitsulmf(7), and sony(7).

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), and export(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, devicetool(1M), libmgr(1M), and robottool(1M). For more information on the GUI tools, see chapter 8, “Graphical User Interface (GUI) Tools”.

Command Options

Many of the commands accept a common set of arguments. These arguments are as follows:

Argument	Meaning
eq	The equipment ordinal of the device being addressed as defined in the mcf file. The device can be either an automated library or a drive.
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 or a partition on a tape. For a disk, the *partition* must be 1 or 2. This is an optional argument for a tape. For a partitioned tape, this argument must be specified, and it must be in the range 0 < *partition* < 256. For a nonpartitioned tape, this argument is 0 by default.
media_type	The media type. For a list of valid media types, see the media(5) man page.
vsn	The volume serial name assigned to the volume.

Some commands accept different 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*

OR

:load media_type.vsn

Note that the first form uses a colon (:) as a separator and the second form uses a period(.).

Terminology

Certain terms are used throughout this chapter. Some of the most commonly used terms, and their meanings, are as follows:

Term	Meaning
Automated library	An automated device for storing tape and optical cartridges.
Cartridge	A tape or magneto optical cartridge.
Partition	A side of a magneto optical disk or a partition on a tape.

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.

Basic Operations

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

- How to start the ASM and ASM-QFS systems
- How to stop the ASM and ASM-QFS systems
- How to turn an automated library on and off
- How to load and unload a cartridge into a drive
- How to label a cartridge
- How to audit a volume
- How to audit an automated library
- How to import a cleaning cartridge
- How to clean a tape drive
- How to clear media errors
- How to remove a cartridge from a stuck drive

How to Start the ASM and ASM-QFS Systems

The sam-initd daemon is started automatically at run level 3 if at least one ASM or ASM-QFS file system is mounted.

You can also start sam-initd without mounting any file systems by entering the following samd(1M) command:

server# samd start

If the sam-initd daemon is already running when the preceding command is entered, the following message is generated:

ASM sam-initd daemon already running

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

How to Stop the ASM AND ASM-QFS System

It is possible to stop sam-initd and leave the ASM or ASM-QFS system mounted. When sam-initd is restarted, pending stages are reissued and archiving is resumed.

If sam-initd is not running, stages are queued but files are not archived until sam-initd is started.

To stop sam-initd, enter the following commands:

server# **samcmd idle** *eq* *(*# see **NOTE)**

server# samd stop

NOTE

The drives in your environment must be idled prior to issuing the **samd stop** command, so enter a **samcmd idle** *eq* command for each *eq* configured in your mcf file. Alternatively, you can also idle the drives by using the samu(1M) operator utility or by using either the robottool(1M) or libmgr(1M) Graphical User Interface (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.

How to Turn an Automated Library on

When an automated library is in the on state, it is under the control of the ASM or ASM-QFS system and can proceed with general operations. 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*

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 chapter 8, "Graphical

User Interface (GUI) Tools", or see the robottool(1M) and libmgr(1M) man pages.

How to Turn an Automated Library off

Placing an automated 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 might turn an automated library off to perform the following tasks:

- You want to stop ASM or ASM-QFS operations for this automated library only.
- You suspect that the automated library or a drive is not functioning properly and you want to run diagnostics.
- You want 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*

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 chapter 8, "Graphical User Interface (GUI) Tools", or see the robottool(1M) and libmgr(1M) man pages.

How to Load a Cartridge into a Drive

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. When you manually load a cartridge, it is generally loaded in the next available drive in the library. If you wish 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).

To manually load a cartridge, use the samu(1M) utility's :load command. This command has the following two possible formats:

:load eq**:**slot[**:**partition]

:load media_type**.**vsn

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 media(5) man page.
partition	A side of a magneto optical disk or a partition on a tape. For a disk, the *partition* must be 1 or 2. This is an optional argument for a tape. For a partitioned tape, this argument must be specified, and it must be in the range 0 < *partition* < 256. For a nonpartitioned tape, this argument is 0 by default.
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 chapter 8, “Graphical User Interface (GUI) Tools”, or see the robottool(1M) and libmgr(1M) man pages.

How 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 has the following format:

:unload *eq*

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 chapter 8, “Graphical User Interface (GUI) Tools”, or see the robottool(1M) and libmgr(1M) man pages.

How to Label 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 automated library uses barcodes, labels = barcodes is set by default. If you want the six low-order digits to become the VSN for the cartridge, include the following line in the /etc/opt/StorageTeksamfs/defaults.conf file:

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 subsections describe these procedures.

WARNING

Labeling and relabeling a cartridge destroys all data on the cartridge. Make sure that this is your intent before labeling or relabeling a cartridge.

How 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 –vsn *new_vsn* **–old** *old_vsn eq***:***slot*[**:***partition*]

new_vsn	The new volume serial name.
old_vsn	The old volume serial 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.
partition	A partition on a tape. This is an optional argument. For a partitioned tape, this argument must be specified, and it must be in the range 0 < *partition* < 256. For a non-partitioned tape, this argument is 0 by default.
media_type	The media type. For a list of valid media types, see the media(5) man page.

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

tplabel -new -vsn *new_vsn eq***:***slot*[**:***partition*]

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

tplabel –old old_vsn **-vsn** new_vsn eq**:**slot[**:**partition]

After issuing the command to label or re-label 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.

These tasks can also be performed using the GUI tools, robottool(1M) and libmgr(1M). For more information on these tools, see chapter 8, "Graphical User Interface (GUI) Tools", or see the robottool(1M) and libmgr(1M) man pages.

How 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 –vsn new_vsn **–old** old_vsn eq**:**slot**:**partition

new_vsn	The new volume serial name.
old_vsn	The old volume serial 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.
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 media(5) man page.

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

odlabel –new –vsn new_vsn eq**:**slot**:**partition

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

odlabel –old old_vsn**–vsn** new_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.

These tasks can also be performed by using the GUI tools, robottool(1M) and libmgr(1M). For more information on these tools, see chapter 8, “Graphical User Interface (GUI) Tools”, or see the robottool(1M) and libmgr(1M) man pages.

How to Audit a Volume

Occasionally the reported space remaining on a tape or optical cartridge may 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*]

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.
partition	A side of a magneto optical disk or a partition on a tape. This is an optional argument for a tape. For a disk, the *partition* must be 1 or 2. For a partitioned tape, this argument must be specified, and it must be in the range 0 < *partition* < 256. For a nonpartitioned tape, this argument is 0 by default.

If the –e option is specified, the remaining space is updated. Otherwise, it is not changed.

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 chapter 9, “The samu(1M) Operator Utility”; chapter 8, “Graphical User Interface (GUI) Tools”; or see one of the following man pages: samu(1M), robottool(1M), or libmgr(1M).

How to Audit an Automated Library (SCSI-attached Only)

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. (See "Automated Library Operations - Systems Without a Mailbox" later in this chapter.)

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*

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 samu(1M) utility's :audit command or by using the GUI tools, robottool(1M) and libmgr(1M). For more information on these tools, see chapter 9, "The samu(1M) Operator Utility"; chapter 8, "Graphical User Interface (GUI) Tools"; or see one of the following man pages: samu(1M), robottool(1M), or libmgr(1M).

NOTE

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

How to Import a Cleaning Cartridge

The ASM and ASM-QFS systems allow you to import a cleaning cartridge to clean the drives. This procedure differs depending on whether or not the cleaning cartridge is barcoded.

Using 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 start with the letters CLN.

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

import *eq*

eq	The equipment ordinal of the automated library 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:

server# 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.

For network-attached automated libraries, the cartridges must be barcoded and you must use the –v option on the import(1M) command, as follows:

server# import –v CLN003 50

The preceding command creates an entry in the library catalog of automated library number 50 for the cleaning cartridge with barcode CLN003. This cartridge must have been physically imported using the library vendor's software.

This procedure can also be performed from within samu 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.

Using Cleaning Cartridges Without Barcodes

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

Import the cartridge using the import(1M) command. The import(1M) command has the following format:

import *eq*

eq	The equipment ordinal of the automated library being addressed as defined in the mcf file.

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:

server# chmed +C 50:77

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

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:

server# chmed –count 20 50:77

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

How to Clean a Tape Drive

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 to be cleaned, 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 using the chmed(1M) command to limit the number of cleaning cycles on cleaning cartridges.

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*

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.

Cleaning tapes are only useful 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 chapter 9, "The samu(1M) Operator Utility"; chapter 8, "Graphical User Interface (GUI) Tools"; or see one of the following man pages: samu(1M), robottool(1M), 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 equals 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

count	The number of cleaning cycles you want to reset.
media_type	The media type. For a list of valid media types, see the media(5) man page.
vsn	The volume serial name assigned to the volume.

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.

How 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.

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

chmed -E media_type**.**vsn

media_type	The media type. For a list of valid media types, see the media(5) man page.
vsn	The volume serial name assigned to the volume.

How to Remove a Cartridge From a Stuck Drive

- If a cartridge becomes stuck in a drive, follow these steps:
- Turn off the drives in the automated library.

- Turn off the automated library.

- Physically remove the cartridge from the drive. Make sure you do not damage either the cartridge or the drive.

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

- If you put the cartridge back into its storage slot, adjust the library catalog to remove the occupied flag for the damaged tape by using the following command:

chmed +o

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.

Automated Library Operations - Systems With a Mailbox

This subsection describes how to import and export media for automated libraries that use a mailbox. Automated libraries that use a mailbox include, for example, the StorageTek 9714, StorageTek 9710, StorageTek 9740, and ADIC Scalar series.

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 export(1M) command moves the cartridge from a storage slot to the mailbox.

If your system does not have a mailbox, the operations described in this subsection do not apply to your system. You should read the next subsection, "Automated Library Operations - Systems Without a Mailbox".

How to Import Cartridges

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

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.).

Manually place the cartridge into the mailbox.

Close the mailbox.

Use the import(1M) command to import the cartridge. This command has the following format:

import *eq*

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.

This step can also be performed from within samu 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.

How to Export Cartridges

To export (eject) cartridges from a media library that uses a mailbox, follow these steps: The cartridge is moved from a storage slot to the mailbox or mail slot.

To move a cartridge from a storage slot to the mailbox, use the export(1M) command in one of the following formats:

export *eq*:*slot*

OR

export media_type.vsn

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 media(5) man page.
vsn	The volume serial name assigned to the volume.

This step can also be performed from within the samu(1M) utility or by using one of the graphical user interface tools, robottool(1M) or libmgr(1M). For more information on these tools, see chapter 9, “The samu(1M) Operator Utility”; chapter 8, “Graphical User Interface (GUI) Tools”; or one of the following man pages: samu(1M), robottool(1M), or libmgr(1M).

Open the mailbox or mail slot using the manufacturer’s suggested operation. There is usually a button near the mailbox.

Automated Library Operations - Systems Without a Mailbox

This subsection describes how to add, remove, or move cartridges if you have an automated library that does not use a mailbox.

Automated libraries that do not have a mailbox cannot import or export cartridges. Therefore, after every manual movement of cartridges into, out of, or within these automated libraries, the library catalog needs to be updated. You can update the library catalog by scanning the barcodes or, if barcodes are unavailable, by auditing the library. For these types of automated libraries, operations differ depending on whether or not the automated library uses barcodes. The following subsections describe these procedures.

Updating the Library Catalog (With Barcodes)

The following procedure describes how to update the library catalog if your system uses barcodes.

Ensure that no archive or stage processes are active by idling all the drives. Use samu(1M) utility's :idle command to idle the drives. The format of this command is as follows:

:idle *eq*

eq	The equipment ordinal of the drive being addressed as defined in the mcf file.

Enter an :idle command for each drive in the automated library.

The drives switch from idle to off when all I/O activity is completed.

Turn off the automated library. You can do this by using the samu(1M) utility's :off command. The format of this command is as follows:

:off *eq*

eq	The equipment ordinal of the automated library being addressed as defined in the mcf file.

Add, remove, or rearrange the cartridges.

Turn on the automated library. You can do this by using the samu(1M) utility's :on command. The format of this command is as follows:

:on *eq*

eq	The equipment ordinal of the automated library being addressed as defined in the mcf file.

After this command has completed, the library state is on.

Turn on the drives. Use the samu(1M) utility's :on command to turn on the drives. The format of this command is as follows:

:on *eq*

eq	The equipment ordinal of the drive being addressed as defined in the mcf file.

Re-initialize the automated library, scan the cartridges, and update the library catalog, which should now contain the VSNs of the cartridges. Use the samu(1M) utility's :audit command to perform this task. The format of this command is as follows:

:audit *eq*

eq	The equipment ordinal of the automated library being addressed as defined in the mcf file.

This entire procedure can also be performed from within the GUI tools, libmgr(1M) or robottool(1M). For more information on these tools, see chapter 9, "The samu(1M) Operator Utility"; chapter 8, "Graphical User Interface (GUI) Tools"; or see one of the following man pages: samu(1M), libmgr(1M), or robottool(1M).

Updating the Library Catalog (Without Barcodes)

The following procedure describes how to update the library catalog if your system does not use barcodes.

Ensure that no archive or stage processes are active by idling all of the drives in the library. Use samu(1M) utility's :idle command to idle the drives. The format of this command is as follows:

:idle *eq*

eq	The equipment ordinal of the drive being addressed as defined in the mcf file.

Enter an :idle command for each drive in the automated library.

The drives switch from idle to off when all I/O activity is completed.

Move the library catalog entries into the historian by using the samu(1M) utility's :unload command. The format of this command is as follows:

:unload *eq*

eq	The equipment ordinal of the automated library being addressed as defined in the mcf file.

Unlock and open the door to the automated library.

Add, remove, or move cartridges within the automated library.

Close the door to the automated library and lock it.

If the automated library's state does not return to on, you must manually set the state to on using the samu(1M) utility's :on command. The format of this command is as follows:

:on *eq*

eq	The equipment ordinal of the drive being addressed as defined in the mcf file.

Turn on the drives that you idled in step 1 of this procedure by using the samu(1M) utility's :on command.

Complete a full audit of the automated library to update the library catalog to reflect the repositioned cartridges. An audit loads each cartridge, reads the label, and updates the library's catalog. From samu(1M), enter the following command, where *eq* is the ordinal of the automated library:

:audit *eq*

eq	The equipment ordinal of the automated library being addressed as defined in the mcf file.

This entire procedure can also be performed from within one of the graphical user interface tools, libmgr(1M) or robottool(1M). For more information on these tools, see chapter 8, "Graphical User Interface (GUI) Tools", or see the libmgr(1M) or robottool(1M) man pages.

Manually Loaded Drive Operations

This subsection describes how to load and unload a cartridge with a manually loaded (or standalone) drive. Each manually loaded drive has its own one-slot library catalog.

How 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 catalog. No further action is necessary.

How to Unload a Cartridge

The following procedure describes how to unload a cartridge from a manually loaded drive.

1) Idle the drive to ensure that no archive or stage processes are active. You can do this using the samu(1M) utility's :idle command. The format of this command is as follows:

:idle *eq*

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 completed.

2) Unload the drive. You can do this using the samu(1M) utility's :unload command. The format of this command is as follows:

:unload *eq*

eq	The equipment ordinal of the drive being addressed as defined in the mcf file.

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.

This entire procedure can also be performed from within the GUI tools, libmgr(1M) or devicetool(1M). For more information on these tools, see chapter 8, "Graphical User Interface (GUI) Tools", or see the libmgr(1M) or devicetool(1M) man pages.

How to View a Library Catalog

To view the library catalog for a manually loaded drive, use the samu(1M) utility's :v command. The format of this command is as follows:

:v *eq*

eq	The equipment ordinal of the drive being addressed as defined in the mcf file.

Chapter 3 - Managing Vendor-specific Automated Libraries

This chapter describes the theory and operations of managing vendor-specific automated libraries. The following topics are presented:

- Automated library hardware and software components
- How to configure automated libraries
- Automated library operations
- Tracking exported cartridges – the historian
- Ampex automated libraries
- ADIC Scalar 224 and Scalar 448 automated libraries
- Sony SCSI-attached B9 and B35 automated libraries
- Sony SCSI-attached 8400 PetaSite automated libraries
- Sony network-attached automated libraries
- StorageTek ACSLS-attached automated libraries
- ADIC/Grau automated libraries
- Fujitsu LMF automated libraries

Automated Library Hardware and Software

An *automated library* is a robotically controlled device designed to load and unload removable cartridges under programmed control without operator intervention. Automated libraries are also known as media changers, jukeboxes, robots, or media libraries.

Cartridges are imported to and exported from the automated library. They are loaded and unloaded automatically or manually. The archiving and staging processes proceed using a site-defined scheme for allocating the number of drives to use.

A *SCSI-attached library* is connected directly to a server using a Small Computer System Interface (SCSI), such as the one for the HP optical automated libraries. The ASM and ASM-QFS systems control these automated libraries directly using the SCSI standard for automated libraries.

The ASM and ASM-QFS systems control a *network-attached automated library*, such as some of the StorageTek, ADIC/Grau, IBM, or Sony automated libraries, by using a software package supplied by the vendor. In these cases, the StorageTek software interfaces with the vendor software using an StorageTek daemon specifically designed for the automated library.

The daemons specific to certain automated libraries are as follows:

Daemon	Description
sam-robotsd	Monitors the execution of robot control daemons. The sam-robotsd daemon is started automatically by sam-initd.
sam-genericd	Controls libraries that conform to SCSI II standards for media changers and the network-controlled ADIC/Grau and Fujitsu LT300 libraries.
sam-stkd	Controls the StorageTek media changers through the ACSAPI interface. This is an optional package available from StorageTek.
sam-ibm3494d	Controls the IBM 3494 tape libraries through the lmcpd interface. This is an optional package available from StorageTek.
sam-sonyd	Controls the Sony network-attached automated libraries through the DZC-8000S interface. This is an optional package available from StorageTek.

The IBM 3494, the StorageTek network-attached automated libraries, Ampex automated libraries, and the Sony network-attached automated libraries require an additional software package to enable them to operate within the ASM and ASM-QFS environments. These packages should be installed when the StorageTek software is installed and configured. To determine which of these packages are installed on your system, enter the following command:

server # pkginfo | grep StorageTek

The preceding command's output shows the StorageTek packages installed on your system. The IBM 3494 package is called LSCibm. The Ampex tape drive package is called LSCdst. The Sony package is called LSCsony.

For more information on these optional packages, see the *ASM Installation and Configuration Guide*. They also require vendor-supplied drivers.

How to Configure Automated Libraries

You can configure automated libraries by editing the Master Configuration File found in /etc/opt/LSCsamfs/mcf. The Additional Parameters field in the mcf file contains 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. Some automated libraries, such as the IBM 3494, the StorageTek network-attached automated libraries, the ADIC/Grau automated libraries, and the Sony network-attached automated 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 StorageTek 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 make sure that you familiarize yourself with the vendor's software.

The mcf File

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

Two important fields must be supplied within the entry for the automated library: the Equipment Identifier field and the Additional Parameters field. These fields differ as follows:

For SCSI-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 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 contains the full path name to the parameters file (described in the next subsection).

For both SCSI and network-attached libraries, the Additional Parameters field is the full path name to the library catalog.

The following mcf entries define a StorageTek network-attached automated library. In this example, two drives are associated with the automated library:

```
# Equipment          Eq        Eq          Family      Dev       Additional
```

# Identifier	Ord	Ty	Set	St	Parameters
#					
/dev/samst/c0t3u0	50	s9	stk	on	
/dev/rmt/3cbn	51	sg	stk	on	
/dev/rmt/0cbn	52	sg	stk	on	

The Parameters File

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), grauaci(7), fujitsulmf(7), and sony(7) man pages for information on the vendor-specific parameters file.

The following example parameters file (/etc/opt/LSCsamfs/stk50) is for a StorageTek automated library. In this example, two drives are described in accordance with the stk(7) man page.

```
#

#  This is file:  /etc/opt/LSCsamfs/stk50

#

hostname = acsls_hostname_here

/dev/rmt/0cbn = (acs=0, lsm=1, panel=0, drive=1)

/dev/rmt/1cbn = (acs=0, lsm=1, panel=0, drive=2)

capacity = (9 = 20971520)
```

The defaults.conf File

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

NOTE

Make sure that there is a space on each side of every equal sign (“ = ”) in the defaults.conf file. Failure to include the spaces causes a syntax error and creates an error message in the ASM or ASM-QFS log file.

Barcodes

If your library 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 follows:

Directive	Action
labels = barcodes	Uses the first part of barcode as the label. Default.
labels = barcodes_low	Uses the last part of barcode as the label.
labels = read	Reads the label from the tape.

If labels = barcodes or barcodes_low, 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 time and the unload wait time for devices using the *dev*_delay and *dev*_unload parameters, respectively. These parameters allows you to set these times to an interval that meets your site’s requirements. These parameters are set in the defaults.conf file.

The format of the *dev*_delay parameter is as follows:

dev_delay = seconds

dev	The device type as specified in the media(5) 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*

dev	The device type as specified in the media(5) 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 parameters, see the defaults.conf(4) man page.

Example

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
labels = barcodes_low
lt_delay = 10
lt_unload = 7
lt_blksize = 256
```

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

Automated Library Operations

A *library catalog* is the central repository of all information that the ASM or ASM-QFS environment needs to find 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 a SCSI-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 SCSI 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 subsections 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.

How 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 the direct SCSI-attached and IBM 3494 automated libraries perform a full audit of the library and build the library catalog. For tape libraries, the audit notes the location and barcode (if available) of all cartridges in the automated library. For optical libraries, the audit involves loading all cartridges and reading the labels. This information is stored in the newly-created library catalog.

However, network-attached StorageTek, ADIC/Grau, Fujitsu, and Sony automated libraries cannot be audited in this manner. For these automated libraries, you can either issue a series of import(1M) commands or you can issue 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.

How 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

-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).

(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
```

How to Import and Export 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 export(1M) commands. You can also perform these tasks using libmgr(1M) or robottool(1M). For more information on importing and exporting cartridges, see chapter 2, “Basic Operations”.

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 export(1M) commands only update the library catalog entries used by the ASM and ASM-QFS systems.

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 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/LSCsamfs/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 only need 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 may 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.

Ampex Automated Libraries

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

The Ampex automated libraries are configured like other SCSI-attached libraries. The tape drivers and runtime library software are supplied by Ampex and must be installed before installing the StorageTek software. The ASM and ASM-QFS systems require version 3.4, or later, of the Ampex DST Tape Device Driver. StorageTek provides a separate package for Ampex support (LSCdst). Install this package following the procedures described in the *ASM Installation and Configuration Guide*.

Configuration

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/LSCsamfs/mcf file. The following example mcf file fragment entries configure a single D2 tape drive with an automated library:

# Equipment	Eq	Eq	Family	Dev	Additional
# Identifier	Ord	Ty	Set	St	Parameters
#					
/dev/samst/c5t6u0	55	rb	am55	on	
/dev/rdst14,1	56	d2	am55	on	/dev/rdst14,7

The first entry is almost identical to the systems configured in the *ASM Installation and Configuration Guide*. This manuals contains an example showing how to configure two automated libraries in a similar manner, using the following procedure:

3) Determine the /dev/samst symbolic links that point to the /devices files with the same Solaris hardware paths as shown in the /var/adm/messages files (in this example, /dev/samst/c5t6u0).

4) Determine an equipment ordinal (55).

5) Define the equipment type (rb).

6) Associate the automated library and drive using a common family set name (am55).

7) Set the device state to on.

8) Specify the location of a library catalog (/usr/tmp/ampx812_cat).

The second entry is for the D2 drive itself. The Equipment Identifier and the Additional Parameters fields are slightly different because the Ampex automated library does not use the naming conventions of the standard Solaris device driver. The first field is the /dev entry for the drive in the format /dev/rdst*a*,*b*. *a* is the number of the drive (14 in the example) and *b* is either 1 (no rewind device in the equipment ID field) or 7 (No I/O device in the additional parameters field). No other value for *b* is accepted by ASM or ASM-QFS systems. As with other drives, these entries are also symbolic links to the hardware path name of the device.

You must also modify /usr/kernel/drv/dst.conf. The DST_ZERO_ON_EW device driver bit must be set as follows:

(set dst_dev_options = 0x00004001)

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).

Formatting a D2 Tape

It is not necessary to use Ampex-supplied commands to format tapes because the ASM system automatically formats them. If desired, though, you can format tapes by using the d2format(1M) command or the Ampex dd2_format_tape utility. These commands are required if you want more than one partition on a tape.

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 popup 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 d2format(1M), 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 chapter 2, “Basic Operations”, to perform importing and exporting.

To export a tape, use the samu(1M) utility’s :unload command or the unload command from libmgr(1M). The unload command positions the tape to a system area and allows the tape to be unloaded from the drive and made available to the operator.

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.

ADIC Scalar 224 and Scalar 448 Automated Libraries

The ADIC 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

The ADIC Scalar 224 and Scalar 448 should not be configured with autoclean or autoload when running StorageTek 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 systems are running.

Operations

Operations on the ADIC automated libraries vary depending on whether a mailbox is available. The mailbox is used to import and export cartridges from the automated library. Those operations that are different when using a mailbox are called out in this subsection.

How to Import Tapes (No Mailbox)

1) To import tapes into an ADIC library without a mailbox, follow these steps:

Turn the automated library state to off by using the set_state(1M) command. The format of this command is as follows:

set_state off *eq*

eq	The equipment ordinal of the device being addressed as defined in the mcf file.

This step can also be performed using samu(1M) utility or by using one of the Graphical User Interface (GUI) tools, robottool(1M) or libmgr(1M). For more information on these tools, see the samu(1M), robottool(1M), or libmgr(1M) man pages.

2) Unlock and open the door to the automated library.

3) Load the cartridges into the available slots.

4) Close and lock the door to the automated library.

At this point, the automated library re-initializes and scans the cartridges it contains. The ASM and ASM-QFS systems update the library catalog, which now contains VSNs of the cartridges. The automated library state displays on.

<u>How to Import Tapes (Mailbox Available)</u>

To import tapes on an ADIC automated library with a mailbox, follow these steps:

1) Enter the import(1M) command. The format of the import(1M) command is as follows:

   import *eq*

eq	The equipment ordinal of the device being addressed as defined in the mcf file.

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

2) Push the OPEN button on the front panel of the library. The door opens.

3) Load cartridges into the mailbox slots.

4) Push the CLOSE button on the front panel of the automated library. Physically close the door to the mailbox.

The automated library moves each cartridge from the mailbox into an empty slot. The library catalog is updated and should now contain the cartridge VSNs.

How to Export Tapes (No Mailbox)

To export cartridges from an ADIC automated library without a mailbox, follow these steps:

1) Turn the automated library state to off by using the samu(1M) utility's :off command. This command has the following format:

   :off *eq*

eq	The equipment ordinal of the device being addressed as defined in the mcf file.

   This step can also be performed using one of the GUI tools, robottool or libmgr. For more information on these tools, see the robottool(1M) or libmgr(1M) man pages.

2) Unlock and open the door to the automated library.

3) Remove cartridges from their respective slots.

4) Close the door to the automated library and lock it.

   The automated library reinitializes and scans the cartridges in the automated library. The ASM and ASM-QFS systems update the library catalog. The removed cartridges are now unavailable in the library catalog. The automated library state displays on.

How to Export Tapes (Mailbox Available)

To export tapes from an ADIC automated library with a mailbox, follow these steps:

1) Enter the export(1M) command. This command has the following two formats:

   export *eq*:*slot*

   OR

   export media_type**.**vsn

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 media(5) man page.
vsn	The volume serial name assigned to the volume.

The automated library moves the tape from the storage slot into the mailbox. The system updates the library catalog.

This step can also be performed using one of the GUI tools, libmgr(1M) or robottool(1M). For more information on these tools, see the libmgr(1M) or robottool(1M) man pages.

2) Push the OPEN button on the front panel of the library. The door opens.

3) Remove the cartridges from the mailbox.

4) Push the CLOSE button on the front panel of the automated library. Physically close the door to the mailbox.

Sony SCSI-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 subsection applies only to Sony SCSI-attached B9 and B35 automated libraries. This information does not pertain to Sony SCSI-attached 8400 PetaSite automated libraries or to any Sony network-attached automated libraries. These other types of libraries are described later in this chapter.

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 “How to Clean a Tape Drive” in chapter 2, “Basic Operations”.

Autoload can be used during the initial loading of cartridges as long as the ASM or ASM-QFS systems are 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. The following subsections describe operations that are different when no mailbox is available.

How to Import a Cartridge without a Mailbox

To import a tape on a Sony B9 or Sony B35 automated library without a mailbox, follow these steps:

1) Invoke the samu(1M) utility, and enter the:unload command. This command has the following format:

   :unload *eq*

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 file.

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 re-initializes and scans the cartridges in the library. The system updates the library catalog, which should now contain the VSNs of the imported cartridges. The automated library state is set to on.

How to Export a Cartridge Without a Mailbox

To export a cartridge on a Sony B9 or Sony B35 automated library without a mailbox, follow these steps:

1) Invoke the samu(1M) utility and enter the :unload command from the command line. This command has the following format:

   :unload *eq*

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 re-initializes and scans the cartridges in the automated library. The system updates the library catalog and the cartridges currently in slots. The removed cartridges are left to the historian file. The automated library state is set to on.

Sony SCSI-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. These mailbox slots can also be used as storage slots. 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 subsection applies only to Sony SCSI-attached 8400 PetaSite automated libraries. This information does not pertain to Sony SCSI-attached B9/B35 automated libraries or to any Sony network-attached automated libraries. These other types of libraries are described elsewhere in this chapter.

How 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) Enable the ASM or ASM-QFS system to recognize the imported cartridges by entering the import(1M) command. The format of this command is as follows:

   import *eq*

eq	The equipment ordinal of the device being addressed as defined in the mcf file.

   This step can also be performed by using one of the GUI tools, libmgr or robottool. For more information on these tools, see the libmgr(1M) or robottool(1M) man pages.

How to Export Tapes

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

How to Export a Tape– Not Using Mailbox Slots as Storage Slots

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

1) Enter 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

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) Enable the ASM or ASM-QFS system to recognize the exported cartridge by entering the export(1M) command. This command has the following format:

   export *eq*

eq	The equipment ordinal of the device being addressed as defined in the mcf file.

   This step can also be performed by using one of the GUI tools, libmgr or robottool. For more information on these tools, see the libmgr(1M) or robottool(1M) man pages.

<u>How to Export a Tape –Using Mailbox Slots as Storage Slots</u>

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) Enable the ASM or ASM-QFS system to recognize the exported cartridge by entering the following command:

   export *eq*

eq	The equipment ordinal of the device being addressed as defined in the mcf file.

   This step can also be performed by using one of the GUI tools, 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 the destination slot is empty.

2) Enter the move(1M) command. This command has the following format:

move eq:source_slot destination_slot

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.

This step can also be performed by using one of the graphical user interface tools, 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 subsection applies only to Sony automated libraries that are network-attached through a Sony DZC-8000S interface. This information does not pertain to Sony automated libraries attached through a SCSI interface. Operations pertaining to Sony SCSI-attached B9 and B35 and to Sony SCSI-attached 8400 PetaSite automated libraries are described earlier in this chapter.

Configuration

The ASM or ASM-QFS configuration process should not be attempted until the Sony automated library is operational and the DZC-8000S software package is installed.

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

The *mcf* File

When creating 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.

The 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 server name. All *keyword*s and *value*s are case sensitive, so they must be entered exactly as used in the configuration file and 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 DZC-8000S server code. This is a required parameter.

- sonydrive drive_id = path. 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. This is a required parameter; there must be one sonydrive line for every drive defined in the mcf file.

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/LSCsamfs/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/LSCsamfs/sonyfile, is the file referenced by the preceding mcf file:

```
#
# This is file: /etc/opt/LSCsamfs/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
```

Building a Catalog

Like a SCSI-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:

  ```
  server# import –v “SEG 99001” 50
  server# import –v vsn2 50
  server# 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 (“).

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

Operations

Chapter 2, “Basic Operations,” 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 from those described in chapter 2, “Basic Operations,” only in the area of importing and exporting cartridges.

Because the physical adding and removing of cartridges in the Sony automated library is done with vendor-supplied utilities, the ASM import(1M) and export(1M) commands and the ASM libmgr(1M) import and export menus affect only the library catalog. The import and export process is described in the following subsections.

Importing 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 import(1M) command to update the library catalog. The syntax for the import(1M) command is as follows:

   import –v [“] *volser* [“] *eq*

" "	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.

Exporting a Cartridge

To export a cartridge, perform the following steps:

1) Use the export(1M) command to remove the entry from the library catalog. The syntax for the export(1M) command is one of the following:

   export *eq*:*slot*

   port media_type.vsn

   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 media(5) man page.

   vsn	The volume serial name assigned to the volume.

   The export(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.

   The import and export procedures can also be performed from within the samu(1M) utility or from within one of the GUI tools, 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 export(1M) commands, see the import(1M) and export(1M) man pages.

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 SCSI-attached automated libraries. However, a StorageTek ACSLS-attached automated library requires additional steps in the installation and configuration procedure as compared to a SCSI-attached automated library.

The ACSLS software package supplied by StorageTek controls the automated library. During the installation of the ASM or ASM-QFS release packages, you must install the StorageTekstk package. This package supplies the stk daemon software that controls the StorageTek automated library through the ACSAPI interface. StorageTek supplies the ACSAPI package. The package is supplied with the FTP package download instructions. To satisfy product dependencies, you must install sampkg first. For more information on this package, see the *ASM Installation and Configuration Guide*.

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

Configuration

The StorageTek software configuration process should not be attempted until the StorageTek ACSLS software package is installed and working. The following subsections 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 SCSI-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 following subsection, “The StorageTek Parameters File”.

- 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.

Example: The following entries in an mcf file define a StorageTek automated library with two associated 9840 drives:

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

The 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 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 SCSI 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 indexes 0-12 that were current at the time of the StorageTek 3.5.0 releases. In general, it is only necessary to supply a capacity entry for an index of new cartridge types or to override the capacity supported by StorageTek.

The defaults are as follows:

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)

device_path_name = (acs = *value*, lsm = *value*, panel = *value*, drive = *value*)

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.

The value specifications are as follows:

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

```
#
# This is file: /etc/opt/StorageTeksamfs/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)
/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/LSCsamfs/examples/ssi.sh. The ssi.sh script is automatically copied to /etc/opt/LSCsamfs/ssi.sh during installation of the samstk package if one does not already exist.

Building a Catalog

Like a SCSI-attached automated library, 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:

  server# **import –v vsn1 50**

  server# **import –v vsn2 50**

  server# **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 then 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:

ASM and ASM-QFS generate the following messages if the samstk package has not been installed. For more information on installing this package, see the *ASM Installation and Configuration Guide*.

May 23 09:12:26 baggins samfs[3171]: /sam1 built May 21 17:51:08 1999.

May 23 09:12:27 baggins robots[3174]: reap_child: stk(50) exited with status 2 (No such file or directory)

May 23 09:12:27 baggins robots[3174]: reap_child: stk will not be restarted

Example 2:

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 3:

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) utitily’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 do not initialize usually indicate a configuration error. Verify that ACSLS is up and running. Verify the hostname. Determine whether you can you ping the hostname 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 4:

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 StorageTek 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

Chapter 2, “Basic Operations”, 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 subsections describe basic operations for the StorageTek ACSLS-attached automated libraries that differ from typical basic operations. These basic operations differ from those described in chapter 2 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 only import and export 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 mailslot are often referred to as the CAP.

When importing and exporting cartridges from an ACSLS-attached automated library, StorageTek commands only affect the StorageTek library catalog. The cartridges are not physically inserted into or removed from the automated library by the StorageTek 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.

How to Import Tapes

To import tape cartridges, use the import(1M) command. This command has the following format:

import –v *vsn eq*

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.

How to Export Tapes Using a Mailbox

Tape cartridges can be exported by slot or by VSN. To export tape cartridges, use the export(1M) command. This command has the following formats:

export *eq*:*slot*

OR

export media_type**.**vsn

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 media(5) man page.
vsn	The volume serial name assigned to the volume.

The StorageTek export(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.

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

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 2.0 interface supplied by ADIC/GRAU. For more information on DAS/ACI, see the *DAS/ACI 2.0 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, the avc (avoid volume contention) and the dismount parameters should both be set to true.

The following subsections 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.

The 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*s and *value*s are case sensitive, so they must be entered exactly as used in the DAS configuration file and 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 different 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/StorageTeksamfs/grau50  50   gr   gr50  -
/var/opt/StorageTeksamfs/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/StorageTeksamfs/grau60   60   gr  gr60  -
/var/opt/StorageTeksamfs/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.

File /etc/opt/StorageTeksamfs/grau50:

```
#
# This is file: /etc/opt/StorageTeksamfs/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
```

File /etc/fs/samfs/grau60:

```
# This is file: /etc/opt/StorageTeksamfs/grau60
#
client = DASclient
server = DAS-server
acidrive DH03 = /dev/samst/c1t1u0
#
# the name "DH03" is from the DAS configuration file
#
```

Building a Catalog

Like a SCSI-attached automated library, 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 two ways of doing this.

- Method 1. Use a series of import(1M) commands. Note that the cartridges must be physically present in the ADIC/Grau 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:

  server# **import –v vsn1 50**
  server# **import –v vsn2 50**
  server# **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 ADIC/Grau 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 ASM or ASM-QFS library catalog slot number, followed by the label, the bar code and then 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 ADIC/Grau automated libraries.

Operations

Chapter 2, "Basic Operations", 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 typical basic operations. These basic operations differ from those described in chapter 2, "Basic Operations", only in the area of importing and exporting cartridges.

Because the physical adding and removing of cartridges in the ADIC/Grau automated library is done with vendor-supplied utilities, the ASM import(1M) and export(1M) commands and the ASM libmgr(1M) import and export menus affect only the library catalog. The import and export process is as described in the following subsections.

Importing a Cartridge

To import a cartridge, perform the following steps:

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

2) The StorageTek import(1M) command to update the library catalog. The syntax for the import(1M) command is as follows:

   import –v *volser eq*

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.

Exporting a Cartridge

To export a cartridge, perform the following steps:

1) Use the StorageTek export(1M) command to remove the entry from the library catalog. The syntax for the export(1M) command is one of the following:

   export *eq*:*slot*

   export media_type.vsn

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 media(5) man page.
vsn	The volume serial name assigned to the volume.

   The StorageTek export(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) ADIC/Grau commands to physically move the cartridge out of the library.

   The import and export procedures can also be performed from within the samu(1M) utility or from within one of the GUI tools, 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 export(1M) commands, see the import(1M) and export(1M) man pages.

Diagnostic Information

Diagnostic information that may be useful when troubleshooting can be found in the following directory:

/var/opt/StorageTeksamfs/.grau

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

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 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 ASM Linkage Operations Guide*; both of these publications are available from the Fujitsu Corporation.

Configuration

The StorageTek software configuration process should not be attempted until the Fujitsu LMF software package is installed and working. The following subsections 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 following subsection describes the Fujitsu parameters file.

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/StorageTeksamfs/*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. Specify the following information for *drivename* and *value*:

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/StorageTeksamfs/lmf50  50  fj  fj50  -
/etc/opt/StorageTeksamfs/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/StorageTeksamfs/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
```

Building 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:

server# **import –v vsn1 50**

server# **import –v vsn2 50**

server# **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 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 then 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

Chapter 2, “Basic Operations”, 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 from those described in chapter 2, “Basic Operations”, only in the area of importing and exporting cartridges.

Because the physical adding and removing of cartridges in the Fujitsu LMF automated library is done with LMF utilities, the StorageTek import(1M) and export(1M) commands and the StorageTek libmgr(1M) import and export menus affect only the library catalog. The import and export process is as follows:

Importing a Cartridge

To import a cartridge, perform the following steps:

1) Fujitsu commands to physically move the cartridge into the library.

2) Use the StorageTek import(1M) command to update the library catalog. The syntax for the import(1M) command is as follows:

   import –v *volser eq*

volser	The *volser* to be added. The fujitsulmf interface verifies that LMF 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.

Exporting a Cartridge

To export a cartridge, perform the following steps:

1) Use the StorageTek export(1M) command to remove the entry from the library catalog. The syntax for the export(1M) command is one of the following:

   export *eq*:*slot*

   export media_type.vsn

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 media(5) man page.
vsn	The volume serial name assigned to the volume.

The export(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.

   The import and export procedures can also be performed from within the samu(1M) utility or from within one of the GUI tools, 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 export(1M) commands, see the import(1M) and export(1M) man pages.

Chapter 4 - The Archiver

Archiving is the process of copying a file from an ASM or ASM-QFS file system to a volume that resides on a removable media cartridge. The ASM and ASM-QFS archiving capabilities include many features, such as those that allow you to specify that files be archived immediately, that files never be archived, and others.

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

The following topics are presented:

- Archiver theory of operations
- archiver.cmd file description
- archiver.cmd directives
- Archiver guidelines
- Troubleshooting the archiver

Archiver – Theory of Operations

The archiver automatically archives ASM and ASM-QFS files. Typically, files are archived to volumes on cartridges that reside in an automated library. Operator intervention is not required to archive and stage the files. Files are archived to a *volume* on a cartridge. Each volume is identified by a unique identifier called a *Volume Serial Name* (VSN). A cartridge can contain one or more volumes. To identify an individual volume, the media type and VSN must be specified.

The archiver starts automatically when an ASM file system is mounted. You can customize the archiver's operations for your site by inserting archiving directives into the following file:

/etc/opt/LSCsamfs/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 time to elapse between complete archiving processes.

The following subsections describe the concept of an archive set and 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 removable media. ASM archived files are compatible with the standard UNIX tar(1) format. This ensures data compatibility with Solaris 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 file system operations. This data consists of directories, symbolic links, the index of segmented files, and removable media information. In the remainder of this section, the term *files* refers to both data files and file system data. The terms *data file* and *file system data* are used only when a distinction is required. The term *file system* refers to a mounted ASM file system. Note that segmented files are enabled through the ASM-Segmentlicense key.

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 directives that apply to all archive sets.

- Archive sets named for each ASM file system are reserved for control structure information. ASM file systems provide a default archive set for each file system. For each file system, both the file system data 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 Process Internals

By default, the archiver makes 1 copy of each archive set, but you can request up to 4 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 different 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. 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 a move(1) command on a file does not cause it to be archived or rearchived. Issuing a move(1) command alters the file's inode, but not the file's 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 appendix B, "Control Structures, Disaster Preparation, and Recovery".

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

You can change the default time references to values far in the past or future, but the archiver adjusts the references so they are always in the following range:

creation_time < time_ref < time_now.

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

Step 1: Scanning for Files to Archive

The archiver scans each file system periodically. The pause between scans is called the *archive interval*. By default, the interval is 10 minutes.

The first time the archiver scans a file system, it descends recursively through the directory tree. It performs a stat(2) on each file to get an inode to examine. 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.

Subsequent scans follow a different pattern. First, the archiver reads the .inodes file. If an inode does not have the archdone flag set, the archiver determines the file name and examines the inode.

After scanning, the archiver performs the following tasks:

1) Determines the archive set to which the file belongs by using the file properties descriptions. If the archive age of the file has been met or exceeded, the archiver adds the file to the archive request for the archive set. The *archive request* is a batch of files that can be archived together. The archive request is sometimes referred to as the *ArchReq*.

   For a segmented file, the segment, not the entire file, is the archivable unit. The properties (for example, the minimum file size) and priorities apply to the individual segments rather than to the entire file. The index of a segmented file contains no user data, so it is treated as a member of the file system archive set. Note that segmented files are enabled through the ASM-Segment license key.

   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.

2) Selects the volumes to be used as the source for the archive copy. This step is performed only if the file is offline. If the file copy is being rearchived, the archiver selects the volume containing the archive copy that is being rearchived.

3) Assigns an archive priority to the file. 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 summation 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 directives in the archiver.cmd file. 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 directive, 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.

4) Adds together all the priorities that apply for a file. The priority of the archive request is set to the highest file priority in the archive request.

Step 2: Composing Archive Requests

After the archiver finishes scanning the file system, it orders the archive requests according to certain default and site-specific criteria. The site-specific criteria allow you to control the order in which files are archived. These criteria, and the order in which they are evaluated, are as follows:

1) If the archive request requires archive sets to be composed according to ownership properties, the archiver separates the archive request according to the file's directory path, user name, or group name. Then the archiver enters all the archive requests into the scheduling queue. This action is controlled by the –reserve directive in the archiver.cmd file.

2) If the archive request requires archive sets to be joined, the archiver groups the files according to the -join *method* directive in the archiver.cmd file. The archiver steps through the archive requests to mark the boundaries in the archive file at which the properties differ. If a –sort *method* is also specified, then the archiver sorts the files within the archive file boundaries according to the -sort *method* and marks the archive request as joined and sorted. If both a join and a sort are specified, the join is performed first. For more information on the –join or –sort directives, see the archiver.cmd(4) man page.

3) If the archive requests are required to be sorted, but not joined, the archiver sorts the files according to the sort method specified on the –sort *method* directive. Depending on the sort method, the archiver tends to keep files together based on the sort method, age, size, or directory location. The archiver marks the archiver request as sorted. By default, the archiver requests are not sorted, so the files are archived in the order in which they are encountered during the file system scan.

4) The archiver determines whether the files are online or offline. It then separates the archive requests into online and offline archive requests. All online files will be archived together, and all offline files will be archived together. The offline files must be staged in and sorted according to their volume.

   Archive copies to be made from 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.

5) If the archiver.cmd file contains the -drives directive to specify multiple drives, the archiver divides the archive request for multiple drives.

   The priority of each archive request created during this process is set to the highest file priority in the archive request. For more information on any of the directives described in the preceding list, see the archiver.cmd(4) man page.

   The archive request queues are written to queue files in the following directory:

   /var/opt/LSCsamfs/archiver/Queues

   These are binary files, and you can display them by using the showqueue(1M) command. For more information on queue files, see Specifying Log Files, later in this chapter.

<u>Step 3: Scheduling from the Queue</u>

When an archive request is ready to be copied, the volumes are assigned to the candidate archive requests as follows:

1) The volume that has most recently been used for the archive set is used if there is enough space on it to hold the archive request.

If an archive request is too big for one volume, it is split into additional archive requests, as needed. Additional volumes available to the archive set are selected in order of occurrence in the catalog.

2) If the archive request is for a single file that is too big to fit on one volume, and it is larger than the ovflmin = *media minimum_size* specification, 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. The ovflmin = directive determines the minimum size of a file that will require more than one volume for archiving. For more information on the ovflmin = directive, see the archiver.cmd(4) man page.

Step 4: Computing the Scheduling Priority

The following scheduling sequence is repeated until all archive requests are processed:

1) 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 of a particular archive request. For more information on customizing the property multiplier, see the –priority directives described on the archiver.cmd(4) man page.

   For each candidate archive request, the archiver 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. You can specify various system resource priorities by using the –priority directive as described on the archiver.cmd(4) man page.

2) Enter each archive request into the scheduling queue. When all archive requests are in the queue, the queue is sorted by the scheduling priority, from highest to lowest, to determine the order in which to assign the archive requests to be copied.

3) Schedule only as many copy operations as drives allowed in an automated library or as allowed by the archive set. Copy operations are performed for more than one archive set at a time. During the copy operations, some archive sets wait in the copy queue until all archive requests are completed.

The scheduling sequence is repeated until all archive requests are processed.

Step 5: Assigning an Archive Request to a Copy Operation

Assigning an archive request to the a copy operation consists of the following steps:

1) Step through each non-joined archive request to mark the archive file boundaries so that each archive file's size is less than the -archmax *target_size*. If a single file is larger than *target_size*, it becomes the only file in an archive file.

   A non-joined archive request is one for which the –join directive is not specified. -archmax *target_size* is the maximum archive file size for an archive set as specified in the –archmax directive. For more information on the –join and –archmax directives, see the archiver.cmd(4) man page.

2) Send the list of VSNs and the list of files to the archiver's copy operation.

Step 6: Completion

As each copy operation is completed, the archiver notes that the archive operation has finished. It releases the drives and volumes that are reserved. When all archive operations have finished, the archiver sleeps for the archive interval.

For each file successfully copied, the following occurs:

- The archive information is entered into the inode.
- If archive logging is enabled, an archive log entry is created.
- An archiver log message is issued if required. 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.

Sample Default Output

The following sample output is from running archiver –l:

Reading "example1.cmd"

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  File system data
   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 process 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.

sam-archiverd is started by sam-initd when ASM or ASM-QFS activity begins. It reads the archiver command file (archiver.cmd) 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. sam-archiverd then monitors sam-arfind and processes signals from an operator or other processes.

Archive Log Files and Event Logging

archiver produces 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. You determine the name of this file. It is not produced by default. For more information on the log file, see the archiver.cmd(4) man page.

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

A *yyyy/mm/dd* 10:59:19 mo OPT003 samfs1.1 1a5.1 samfs1 27.1 4096 big/HiRes d 0

A *yyyy/mm/dd* 10:59:19 mo OPT003 samfs1.1 1a5.a samfs1 34.1 4096 big/QT/Srs d 0

A *yyyy/mm/dd* 10:59:20 ib E00000 all.1 110a.1 samfs1 20.5 14971 myfile f 0

Reading left to right, the fields in the previous listing have the following content:

Field	Content
1	Archive activity. A for archived. R for rearchived. U for unarchived.
2	Date of archive action in *yyyy*/*mm*/*dd* format.
3	Time of archive action in *hh*:*mm*:*ss* format.
4	Archive media type. For information on media types, see the media(5) man page.
5	VSN.
6	Archive set and copy number.
7	Physical position of start of archive file on media (tar(1) file) and file offset on 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 written on only 1 volume. Length of section (if file is written on multiple volumes).
11	Name of file.
12	Type of file. d for directory. f for regular file. l for symbolic link. R for removable media file. I for segment index. S for data segment. Segmented file features are enabled through the ASM-Segmentlicense

Field	Content
	key.
13	Section of an overflowed file or segment. Segmented file features are enabled through the ASM-Segmentlicense key.

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

The archiver.cmd File Description

By default, the archiver runs whenever sam-initd is started and an ASM 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/LSCsamfs/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 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 as a suffix to the number that signifies the unit:

Letter 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.

NOTE

Make sure that there is a space on each side of every equal sign (=) in the archiver.cmd file. Failure to include the spaces causes a syntax error, prevents the archiver from running, and writes an error message to the log file for the ASM or ASM-QFS system.

An archiver.cmd File Example

The following example shows a sample archiver.cmd file. The comments on the right hand side indicate the different types of directives as listed in the previous subsection.

```
interval = 30m                              # General directives
```

```
logfile = /var/opt/LSCsamfs/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
images.1 -drives 2 -join path -sort size
samfs1_all.1 -drives 2
samfs2_all.1 -drives 2
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 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 complete. It then reads the modified archiver.cmd file and restarts.

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

*** No archiving will be performed. ***

Then the archiver waits to be restarted, which occurs when the archiver.cmd file changes or when the archiver receives a SIGINT or :arrestart.

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 –A

The preceding command lists all options and writes a listing of the archiver.cmd file, VSNs, 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/LSCsamfs/archiver.cmd. If you run 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 for a complete description of this command.

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

archiver.cmd Directives

The following subsections introduce the archiver.cmd directives.

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 archiver executes periodically to examine the status of all mounted ASM file systems. The timing is controlled by the archive interval. The *archive interval* is the time between complete archive operations (scanning and copying to volumes) on each file system. To change the time, use the following directive:

interval = *time*

The default interval is 10 minutes. If the archiver receives a SIGALRM from sam-initd or from the samu(1M) utility's :arrun command, it begins scanning all file systems immediately. For more information on specifying the archive interval, see the archiver.cmd(4) man page.

Specifying Log Files

The archiver can write to an event-tracing log file. The archiver has an extensive action tracing facility. To direct archiver messages to a file, use the following directive:

trace = *pathname* [$event_1$ $event_2$ ...]

For *pathname*, specify the absolute path name to the trace log file.

Additional arguments to the directive determine the events that are traced. This facility is intended for diagnostic purposes. For more information on the tracing log file, see the archiver.cmd(4) man page.

Specifying a Directory for Queue and Data Files

By default, the queue files and data files are written to the following directory:

/var/opt/LSCsamfs/archiver

You can specify a different directory by using the datadir directive within the archiver.cmd file. Set the name of the directory used for the archiver's data files to *dirname*, as follows:

datadir = *dirname*

Except for ReservedVSNs, the data files are used only by the archiver.

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 following directive:

drives = auto_lib count

In the preceding syntax, *auto_lib* is the family set name of the automated library as defined in the mcf file, and *count* is the number of drives to be used for archiving activities.

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.

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 and 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% of the media capacity. For example, the following archmax directive can be used for a 20-gigabyte tape:

archmax = sg 1G

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

To change the defaults, use the following directive:

archmax = media target_size

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

Controlling the Archive Buffer Size

The bufsize directive sets the archive buffer size for a specific tape media type to *buffer_size* multiplied by *dev_blksize*. The *dev_blksize* value is based on the default buffer type for a tape. The format of this directive is as follows:

bufsize = *media buffer_size* [lock]

For *media*, specify a media type as specified on the media(5) man page.

For *buffer_size*, specify an integer in the following range: 2 < *buffer_size* < 32. The default *buffer_size* is 4.

If lock is specified, the archiver locks the archive buffer.

The right amount of buffers keeps the archive process flowing. If your tapes stop frequently while archive files are being written to them, you might want to increase the *buffer_size*. A buffer size that is too large, however, uses up more memory than is necessary.

Controlling Volume Overflow

Volume overflow is the process of allowing archived files to span multiple volumes. For more information on volume overflow, see chapter 11, “Advanced Features”.

Before using volume overflow, make sure that you understand the concept. Use volume overflow with caution only after thoroughly assessing the affect 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, ovflmin is set to 8 exabytes (2^{63} bytes, which is effectively disabled). To change this default, use the following directive:

ovflmin = *media minimum_file_size*

For *media,* specify the media type as defined on the media(5) man page. For *minimum_file_size*, specify the minimum size of the file to overflow.

<u>Example</u>: Assume that many files exist with a length that is a significant fraction (say 25%) of the mo medium. These files partially fill the volumes and leave unused space on each volume. To get better packing of the volumes, set ovflmin for the mo medium 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. ovflmin can also be set for an individual archive set.

Delaying Archiver Startup

The wait directive causes the archiver to wait for a SIGUSR1 signal. When this signal is received, normal archiver operations are begun. By default, the archiver begins archiving when started by sam-initd(1M). To delay archiving, use the following directive:

wait

Delay Time

The delay directive specifies the time delay before starting the archiver scan. You can use this time delay to wait for all automated libraries to initialize. The format of this directive is as follows:

delay = *time*

delay = 3m

Rename the Event Notification Script

The notify directive sets the name of the archiver's event notification script file to *filename*. The format of this directive is as follows:

notify = *filename*

The default filename is as follows:

/opt/LSCsamfs/sbin/ar_notify.sh

This file is executed by the archiver to allow you to process various events in a site-specific fashion. 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.

Rename the Directory Used for Archiver Queue Files

The queuedir directive specifies the directory to contain the archiver's queue files. The format of this directive is as follows:

queuedir = *dirname*

The default data directory is as follows:

/var/opt/LSCsamfs/archiver/Queues

The queue files are used only during archiver execution.

Controlling 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.

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 following directive:

fs = *fsname*

For *fsname*, specify the file system name as defined in the /etc/opt/LSCsamfs/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.

Specifying A General Log File

The archiver produces 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 following directive:

logfile = *pathname*

For *pathname*, specify the absolute path name to the log file. By default, this file is not produced.

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_criteria*$_1$ *search_criteria*$_2$ **…**]

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*. For more information on the no_archive *archive_set_name*, see the "Preventing Archiving" subsection later in this chapter.
path	A path relative to the mount point of the file system. This allows an archive set membership directive to apply to multiple ASM 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 subsections.

Examples: The following are typical archive set membership directives:

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

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 following letters:

Letter	Meaning
b	Bytes

Letter	Meaning
k	Kilobytes
M	Megabytes
G	Gigabytes
T	Terabytes

Examples:

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

This example specifies that all files at least 500 kilobytes in size, but less than 100 megabytes, belong to the archive set big_files. Files bigger than 100 megabytes belong to the archive set huge_files.

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.

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.

Release and Stage search criteria: -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 following syntax is used to manipulate these file attributes:

-release *attributes*

-stage *attributes*

The *attributes* for the -release directive follow the same conventions as the release(1) command, as follows:

Attribute	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 *attributes* for the -stage directive follow the same conventions as the stage(1) command, as follows:

Attribute	Meaning
a	Associative stage the file.
n	Never stage the file.

Example: 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

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. -name *regex* specifies that any complete path matching the regular expression *regex* is a member of the archive set.

regex 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.

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

images . -name \.gif$

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

satellite . -name ^GEO

Example 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$
```

Example 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

Example 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 (/). 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

Preventing Archiving

Archiving can be prevented by including the files in the archive set no_archive.

Example: 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:

fs = samfs1

no_archive tmp

no_archive . –name .*/tmp/

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 may 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.

Example. Assume 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, since 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 would 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 subsections describe the archive copy directive arguments.

Releasing Disk Space After Archiving: *-release*

The disk space for files can be automatically released after an archive copy is made by placing the following specification after the copy number:

-release

Example:
ex_set . -group images
1 -release 10m

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

Delaying Disk Space Release: *-norelease*

You may not wish to release disk space until multiple archive copies are completed. The
-norelease specification prevents the automatic release of disk cache until all copies marked with -norelease are made.

Example: 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

The archive age for files can be set by specifying the archive age as the next field on the directive. The archive age can be specified with the following suffix characters:

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

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

ex_set data

1 1h

If the archive age is not specified, the default is 4 minutes.

Automatic Unarchiving

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 different media using different archive ages.

Example: This 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, a second and third archive copy is made. The first copy is then unarchived.

For more ways to control unarchiving, see "Controlling Unarchiving" later in this section.

More than One Copy for File System Data

If more than one copy of file system data is required, copy definitions can be placed in the directive file immediately after an fs = directive.

Example:

fs = samfs7

1 4m

2 1h

In this example, copy 1 of the file system data for the samfs7 file system is made after four minutes and a second copy is made after one hour.

Archive Set Processing Directives

This subsection 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:

archive_set_name.*copy_number* [-*param*$_1$ -*param*$_2$ …]

The sort methods and priority controls can be difficult to manage for a large number of archive sets. 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. Directives set for individual archive set copies override directives set by allsets directives. For more information on the allsets archive set, see the archiver.cmd(4) man page.

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. This is specified by using the -drives directive.

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* directive in conjunction with the–drives directive to set the minimum size for splitting archive requests among drives. For example, you can use the –drivemin *min_size* directive 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 *min_size* directive is the setting for the –archmax *target_size* directive. The –archmax *target_size* directive's default setting is the *target_size* for the specific volume being used.

To display the setting for -drivemin, enter the following command:

server# archiver -lv

The -drivemin directive has the following format:

–drivemin *min_size*

The –drivemin *min_size* directive sets the minimum size of the multiple drives for the archive set to *min_size*. When you use the –drives directive, 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 directive.

An archive request is evaluated against both the -drives and -drivemin directives, 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.

If *min_size* is zero, an attempt is made to split for the full number of drives specified.

Example. Assume that you are splitting an archive set named big_files over 5 drives. Depending on its size, this archive set could be split as follows:

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/LSCsamfs/archiver.cmd file for this example:

params

bigfiles.1 –drives 5 –drivemin 10G

endparams

Associative Archiving

Associative archiving is used if you specify the –join path directive. 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 directive 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 may 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 directive 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 wish 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 may wish to access those images together at one time. These static images can be more efficiently accessed if you archive them contiguously based upon their directory location. For example:

patient_images.1 -join path

NOTE

The -join path directive 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 directive 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 directive 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 options are mutually exclusive. To sort an archive set, use the -sort directive with the argument age or size, as follows:

cardiac.1 –sort path

cardiac.2 -sort age

catscans.3 -sort size

The first example forces the archiver to sort an archive request by path name. The second example forces the archiver to sort an archive set copy called cardiac.2 by the age of the file, youngest to oldest. The third example 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, not accessed data, 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 4 weeks. During this time, all of this patient's files are on fast media (copy 1=mo). After 4 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 6 months for a re-check, the first access of the data is from tape (copy 2). Now, the archiver automatically creates a new copy 1 on disk to assure 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 directive directs the archiver to write only the initial tape mark. In addition, if the –tapenonstop directive is specified, the archiver enters the archive information at the end of the copy operation.

For more information on the –tapenonstop directive, 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 directive 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 directive reserves volumes for an archive set. When the –reserve directive 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 directive 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 directive for one, two, or all three possible forms. The three forms can be combined and used together in an archive set directive definition. For example, the following directive definition creates a reserved name based upon an archive set, a group, and the file system:

  params

  tracefiles.1 lt -reserve set -reserve group -reserve fs

  endparams

  The information regarding reserved volumes is stored in the following file:

  /var/opt/LSCsamfs/.archive/ReservedVSNs

  Note that the lines within the ReservedVSNs file contain the media type, the VSN, the reserve information, and the reservation date and the 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. The following fragment is from an example ReservedVSNs file:

  mo OPT001 hgmaux.1// 2000/05/13 20:07:29
  mo OPT001 samfs9.1// 2000/05/24 10:00:26

  mo OPT001 samfs9.1// 2000/05/26 17:20:29
  lt SLOT2 hgm1.1// 2000/05/29 15:56:11

  As the preceding example shows, 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. For more information on this file, see the ReservedVSNs(4) man page.

You can display the reserve information by using the following command:

server# archiver –lv

The following paragraphs illustrate each form showing the directive, 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 follows:

Directive and Keyword	Reserved Name Examples
-reserve set	users.1//
	Data.1//

- 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. dir uses the directory path component immediately following the path specification of the archive set definition. user and group are self-explanatory.

Directive and Keyword	Reserved Name Examples
-reserve dir	proj.1/p105/
	proj.1/p104/
-reserve user	users.1/user5/
	users.1/user4/
-reserve group	data.1/engineering/

NOTE

The -reserve directive 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 ReservedName.

Directive and Keyword	Reserved Name Examples
-reserve fs	proj.1/p103/samfs1
	proj.1/p104/samfs1

The -reserve options can be set globally using a general directive. The syntax for the directive follows:

reserve = [dir | user | group]

reserve = fs

A complete archive example using reserved volumes is presented at the end of this chapter as Example 4.

When the archiver initializes and reserves volumes to archive sets, it reads the ReservedVSNs file to determine previous volume reservations. If this file is not found, a warning message is sent to the system log.

While reading the ReservedVSNs file, the archiver edits it to remove badly formatted lines and reservations for volumes that have been relabeled. A relabeled volume is identified by a label time that is more recent than the reservation time, such as volumes that have been relabeled as part of recycling. Lines referring to previous archive set reservations are commented out.

If a reserved volumes backup file (/var/opt/LSCsamfs/archiver/ReservedVSNs.bak) does not exist, the current reserved volumes file becomes the backup. The edited file becomes the new reserved volumes file.

Media is automatically repooled when it is relabeled, and entries for volumes that no longer exist (because they are relabeled with a different label) are cleaned up. The archiver maintains the ReservedVSNs file. When it recognizes that the label date on media is older than the reservation date, it removes the entry from the ReservedVSNs file and allows the volume to be re-reserved.

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.

The following table lists the 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 achiver.cmd file assigns volumes to archive sets. This section starts with a vsns directive and ends with an endvsns directive.

Collections of VSNs are assigned to archive sets by directives of the following form:

archive_set_name.*copy_number media_type vsn_expression* ... [-pool *vsn pool_name* ...]

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 *media_type* is the media mnemonic as defined in the media(5) man page.

The volumes are noted by one or more *vsn_expression* keywords, which are regular expressions as described in 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.

Example 1. 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]

Example 2. 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, StorageTek recommends that you do not assign a VSN to more than one archive set.

NOTE

Make sure you assign volumes to the archive set for the file system data 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 file system data, see the samfsdump(1M) man page or see appendix B, “Control Structures, Disaster Preparation, and Recovery”.

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” in the immediately preceding subsection.

As volumes are selected for use by an archive set, they are assigned a reserve name using the archive set component. For more information on reserving volumes, see the “Reserving VSNs” subsection previously in this chapter.

After 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. The VSN pools can be viewed by entering the following command:

```
server# archiver -lv
```

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

The *vsn_pool_name* identifies the pool. The *media_type* is the 2-character media identifier as defined in the media(5) man page. The *vsn_expression* is a regular expression as defined in the regcmp(3G) man page; there can be one or more *vsn_expression* keywords.

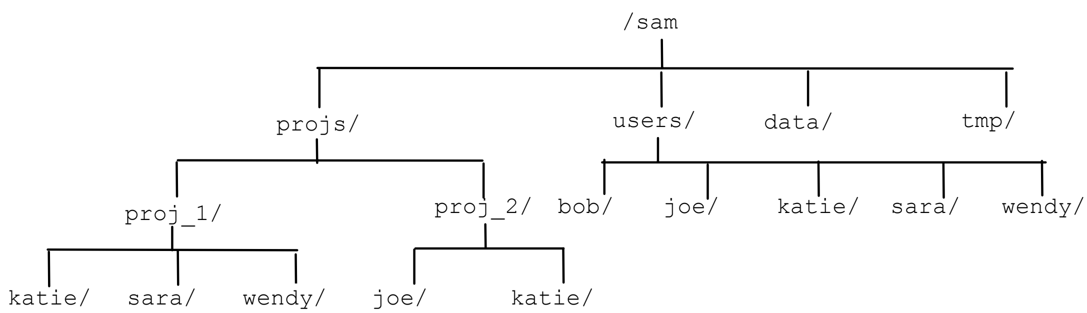
Example. The following example uses four VSN pools: users_pool, data_pool, proj_pool, and scratch_pool. If one of the three specific pools is out of VSNs, the scratch pool VSNs are selected. The example is as follows:

```
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

Archiver Examples

All archiver examples in this subsection assume the following 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 6 cartridges. The example shows the output produced by the following command:

server# **archiver -lv**

The following output shows that the default media selected by the archiver is type mo. Only the mo media are available:

Data directory: /var/opt/LSCsamfs/archiver
Notify file: /opt/LSCsamfs/sbin/ar_notify.sh
Trace output to: /var/opt/LSCsamfs/archiver/trace

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 VSNs, 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 file system data 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 File system data
    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 VSNs in the indicated order:

```
Archive sets:
allsets

samfs.1
 media: mo (by default)
 VSNs:
   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 file system data. There is a manually mounted DLT tape drive in addition to the optical automated library from Example 1. 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:

server# **archiver –lv 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/LSCsamfs/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]       # File system data 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.

```
Data directory: /var/opt/LSCsamfs/archiver
Notify file: /opt/LSCsamfs/sbin/ar_notify.sh
Trace output to: /var/opt/LSCsamfs/archiver/trace

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-------
```

Here is the organization of the file system. Files bigger than 512000 bytes (500 kilobytes) are archived after four minutes; all other files are archived after thirty seconds.

```
Archive file selections:
Filesystem samfs  Logfile: /var/opt/LSCsamfs/archiver/log
samfs File system data
    copy:1  arch_age:240
big  path:. minsize:502.0k
    copy:1  arch_age:240
all  path:.
    copy:1  arch_age:30
```

Note the division of the archive sets among the removable media.

```
Archive sets:
allsets

all.1
 media: mo
 VSNs:
   optic03
   optic04
   optic10
   optic12
```

```
   optic13
   optic20
   optic21
   optic22
   optic23
 Total space available:    6.3G

big.1
 media: lt
 VSNs:
   TAPE01
   TAPE02
   TAPE03
   TAPE04
   TAPE05
   TAPE06
 Total space available:   42.8G

samfs.1
 media: mo
 VSNs:
   optic00
   optic01
   optic02
 Total space available:    2.6G
```

Example 3

In this example, user files and project data files are archived to different 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 VSNs. Files in the directories tmp and users/bob are not archived. Archiving is performed on a 15-minute interval and an archival record kept.

The example shows the output from the following command:

server# **archiver -lv example3.cmd**

```
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/LSCsamfs/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$
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

Data directory: /var/opt/LSCsamfs/archiver
Notify file: /opt/LSCsamfs/sbin/ar_notify.sh
Trace output to: /var/opt/LSCsamfs/archiver/trace

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-------

Archive file selections:
Filesystem samfs  Logfile: /var/opt/LSCsamfs/archiver/log
samfs File system data
    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


Archive sets:
allsets

joe.1
```

```
 media: mo
 VSNs:
   optic01
 Total space available:  934.2M

joe.2
 media: lt
 VSNs:
   TAPE01
   TAPE02
 Total space available:   14.7G

pict.1
 media: mo
 VSNs:
   optic02
 Total space available:  781.7M

pict.2
 media: mo
 VSNs:
   optic03
 Total space available:    1.1G

prod.1
 media: mo
 VSNs:
   optic20
   optic21
   optic22
   optic23
 Total space available:    3.3G

prod_big.1
 media: lt drives:2
 VSNs:
   TAPE03
   TAPE04
 Total space available:   12.1G

prod_big.2
 media: lt drives:2
 VSNs:
   TAPE05
   TAPE06
 Total space available:   16.0G
```

```
proj_1.1
 media: mo
 VSNs:
   optic10
 Total space available:   85.5M

proj_1.2
 media:

   optic12
   optic13
 Total space available:  981.9M

samfs.1
 media: mo
 VSNs:
   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 this example does not use the directory structure pictured at the beginning of the examples.

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 archive log is kept.

The following example shows the 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/LSCsamfs/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]$
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

Data directory: /var/opt/LSCsamfs/archiver
Notify file: /opt/LSCsamfs/sbin/ar_notify.sh
Trace output to: /var/opt/LSCsamfs/archiver/trace

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/LSCsamfs/archiver/log
samfs File system data
    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 VSNs:
   optic10
 Total space available:   85.5M

proj_pool media: mo VSNs:
   optic12
   optic13
 Total space available:  981.9M

scratch_pool media: mo VSNs:
   optic20
   optic21
   optic22
   optic23
 Total space available:    3.3G

users_pool media: mo VSNs:
   optic01
   optic02
   optic03
 Total space available:    2.7G

Archive sets:
allsets

data.1
  reserve:/group/
 media: mo
 VSNs:
```

```
  optic10
  optic20
  optic21
  optic22
  optic23
 Total space available:    3.4G

proj.1
  reserve:/dir/fs
 media: mo
 VSNs:
   optic12
   optic13
   optic20
   optic21
   optic22
   optic23
 Total space available:    4.2G

samfs.1
 media: mo
 VSNs:
   optic00
 Total space available:  939.7M

users.1
  reserve:/user/
 media: mo
 VSNs:
   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 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 ASM's experiences. 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.

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) Archive interval. The archive interval should be based upon how often files are created and modified, and whether your want to save all modification copies saved. Remember, the archive interval is the dead time between finding archive copies and copying them to media. A very short archive interval keeps the archiver running almost continuously.

4) File systems. Multiple ASM file systems generally increase the performance of the archiver as compared to a single ASM file system. The archiver is a multithreaded program designed to run in a parallel fashion. Multiple file systems can be scanned in considerably less time than a single file system.

5) Directory usage. You should use directory structures to organize your files within the ASM file system, like UNIX file systems. For performance considerations, ASM recommends that you do not place more than 2000 files in a directory.

6) Multiple archive copies. You should 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) File system data. The file system data (directory structure, file names, and so on) are 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 need to restore with the .inodes file using the sammkfs -r command, keeping file system data together on a single volume speeds up the recovery. If you do not wish 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 file system data, see appendix B, “Control Structures, Disaster Preparation, and Recovery”.

Troubleshooting the Archiver

Upon initial setup, the archiver may 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:

  *** No Archiving Will Take Place ***

  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 on a regular basis 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 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 may 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 VSNs 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/LSCsamfs/archive to handle the exceptions of no space or no VSNs 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

Here is a checklist of reasons why your ASM or ASM-QFS environment might not be archiving files:

- archiver.cmd file has a syntax error. Run the archiver -lv command to identify the error, then correct the flagged lines.
- 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.
- No volumes are available. This is also shown with the archiver –lv command. Add more volumes as needed. You may have to export existing cartridges to free up slots in the automated library.
- The volumes for an archive set are full. You can export cartridges and replace them with new cartridges (make sure that the new cartridges

are labeled), or you can recycle the cartridges (see chapter 7, “The Recycler”).

- The VSN section of the archiver.cmd file fails to list correct media. Check your regular expressions and VSN pools to ensure that they are correctly defined.

- There is not enough space to archive any file on the available volumes. If you have larger files and it appears that the volumes are nearly full, the cartridges might be as full as the ASM or ASM-QFS environment allows. If this is the case, add cartridges or recycle.

- The archiver.cmd file has the no_archive directive set for directories or file systems that contain large files.

- The archive -n (archive never) command has been used to set too many directories and files never archive.

- Large files are busy. Thus, they never reach their archive age and are not archived.

- Hardware or configuration problems with the automated library.

Additional Archiver Diagnostics

In addition to the previous list, you should check the following when troubleshooting the archiver:

1) Check the syslog file (by default, /var/adm/sam-log) for any archiver messages that can indicate the source of the problem.

2) Assure 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.

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.

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.

Additional common reasons for why an ASM or ASM-QFS environment is not releasing files are as follows:

- Files can only be released after they are archived. Check one of the previous lists, “Why Files Are Not Archiving”.

- The archiver.cmd file has the -norelease directive set for too many archive copies.

- The release -n command has been used to flag directories and files to prevent release.

- The archiver.cmd file has the -release n directive set for too many directories and files.

- The releaser high water mark is set too high, and automatic releasing occurs too late. Verify this in the samu(1M) utility’s m display or with libmgr(1M). Lower this value.

- The releaser low water mark 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, be archived, and be released.

For more information on why files are not being released, see chapter 6, “The Releaser”.

Chapter 5 - The Stager

This chapter describes the ASM and ASM-QFS file staging capability. Typically, *staging* is the process of copying a nearline or offline file from its archive storage back to online storage. The staging capabilities allow you to stage files immediately, to never stage files, 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 contains the following topics:

- Logging file staging activities
- The preview.cmd file - prioritizing preview requests

Logging File Staging Activities

You can request that the ASM or ASM-QFS file system collect file staging event information and write it to a log file. To begin logging file staging events, enter a stage directive in the samlogd.cmd file. This directive has the following format:

stage = filename [event]

For *filename*, specify a full path name. For *event*, specify zero or more staging activities to log: start, finish, cancel, or all (default).

The resulting log file can contain information on the event type, the media type, the volume serial name (VSN), and other aspects of staging a file. For more information on the log file, see the sam-logd(1M) man page.

<u>Example</u>: The following directive line in file samlog.cmd writes information regarding the beginning of each stage to file /tmp/stage.log:

stage = /tmp/stage.log start

The preview.cmd File - Prioritizing Preview Requests

Archive and stage requests that cannot be immediately satisfied go to the preview queue for future consideration and become preview requests. By default, preview requests are satisfied in First In First Out (FIFO) order.

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/LSCsamfs/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 water mark (HWM) or low water mark (LWM) settings.

The preview directives are read by sam-initd at start up, and the specified values are stored in shared memory. The directives must be listed one per line. Changes made to this file while sam-initd is running do not take effect until sam-initd 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.

NOTE

Make sure that there is a space on each side of every equal sign (=) in the preview.cmd file. Failure to include the spaces causes a syntax error, prevents the previewer from running, and creates an error message in the ASM or ASM-QFS log file.

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.

- File system specific directives, 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 specific 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 system specific 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 file system specific directives. That is, they must appear prior to any fs = directives.

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. 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.

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.

Water Mark Directives (Global or File System Specific)

The water mark preview request directives can be used as either global or file system specific directives. The water mark priority directives determine the water mark 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 the subsection “Enforcing Stage Requests”. This subsection appears a few pages later.

The four water mark priority settings are as follows:

lwm_priority = *value*

The value by which the wm_priority factor changes for archiving requests when the file system is below the LWM level. The default lwm_priority is 0.0.

lhwm_priority = *value*

The value 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 the the file system is filling up. The default lhwm_priority is 0.0.

hlwm_priority = *value*

The value 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 hlwm_priority is 0.0.

hwm_priority = *value*

The value by which the wm_priority factor changes for archiving requests when the file system is above the HWM level. The default hwm_priority is 0.0.

Together, the four water mark 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 water mark priority setting, with or without the chmed(1M) command's p option.

The water mark priorities are used only to calculate media requests for archiving; they are not used to calculate media requests for staging.

Example. 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 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:

```
priority = vsn_priority + wm_priority + (age_priority *
time_in_sec_as_preview_request)
```

How to Set Up a Preview Request Priority Scheme

It is only necessary to change the default preview request FIFO scheme 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 archival requests.

- Condition 2: Ensure that archival 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 above scenarios and influence the configured ASM or ASM-QFS environment.

Since 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, and the priorities of each preview request.

Example. The following is a sample preview.cmd file that 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 archival requests. This example assumes the following:

Several different requests are sitting in the queue for 100 seconds.

The default vsn_priority is 1000.

The total request priorities are calculated as follows:

- Archive VSN w/priority, LWM: 1000 + (-200) + (1 x 100) = 900
- Stage VSN w/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 shadowed files to lower the percent full of the file system, completing the pending archival requests is the next best chance to keep the file system from filling up. *Shadowed files* are those with at least one archive copy but are still resident in the disk cache.)

In this situation, the preview.cmd file can be as simple as:

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 easily configure the preview.cmd file with the following directive to give the user and their media the appropriate priority for media drives:

vsn_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 de-prioritize the user's media, do a 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 archival or stage requests one over the other. In this case, the affected directives are:

  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, archival 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

Chapter 6 - The Releaser

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 only release 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.

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 trigger the staging of the rest of the file back to disk cache from the archival media. These features, and many others, are described in this chapter.

- This chapter contains the following topics:
- Releaser overview
- Definitions
- releaser.cmd file
- Releaser operation
- How to configure the releaser for files in cache
- Running the releaser manually
- Troubleshooting the releaser

Releaser Overview

When the disk cache utilization for a file system exceeds its configured high water mark, the file system sends a message to sam-initd to invoke the releaser. First, the releaser reads its command 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 water mark. Under normal conditions, 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.

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 water mark. 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 water mark. 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 all remaining online files are marked release -n. The ASM and ASM-QFS file systems start the releaser again later.

Definitions

The following terms are used throughout this chapter:

- Age. Age is 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, a file with a modify time of 10:10 has a data modified age of 5 minutes. For more information on the sls command, see the sls(1) man page.

- Candidate. A candidate is a file that is eligible to be released. The following list contains reasons why a file would not be a candidate:

  The file is already offline.

  The file has not been archived.

  The archive command file specifies the -norelease attribute for the file, and all copies have not yet been made.

  The file is marked 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 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 10 minutes ago or less.

  The file is flagged release –p (partial release) and is already partially released.

- Priority. The priority is the sum of two types of priority: age priority and size 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. Candidate files with numerically larger priorities are released before candidates with numerically smaller priorities.

- Weight. The weight is a numeric value that biases the priority calculation to include file attributes in which you are interested and 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.

releaser.cmd File

The releaser.cmd file consists of lines of text that are either comment lines, denoted by a leading pound sign (#) or directive lines. The directive lines perform the following functions:

- Set the parameters used to calculate a priority for each file. This is the most important function.
- Set the location of log files.
- Put the releaser into debug mode. Files are not released in debug mode.

NOTE

Make sure that there is a space on each side of every equal sign (=) in the releaser.cmd file. Failure to include the spaces causes a syntax error, prevents the releaser from running, and creates an error message in the log file.

For more information on the releaser.cmd file, see the releaser.cmd(4) man page. The following subsections describe the directives that can be specified in the releaser.cmd file.

Releaser logfile Directive

If a logfile = *filename* directive is specified in the releaser.cmd file, the releaser appends its activity to the indicated file name, or the releaser creates the file name if it does not exist. 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 2002
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 2002) 10001 min, 500 blks
/sam1/testdir0/filevp
10500 (R: Wed Apr 21 18:48:10 CDT 2002) 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 April 28, 2002
```

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

Release Priority Directives

The releaser calculates the release priority of each file in a file system by multiplying various weights by the corresponding file properties and then summing the results. The weights are floating-point values between 0.0 and 1.0. The default value for weights is 1.0.

The following directives specify the priority weights used to calculate the release priority for a file:

weight_age = *float*

weight_age_access = *float*

weight_age_modify = *float*

weight_age_residence = *float*

weight_size = *float*

Two priorities are used to calculate the final release priority for a file: the age of the file and the size of a file.

After the age and size priorities are calculated, they are summed to yield the final release priority for the file, as follows

release_priority = age_related_priority + size_related_priority

Age of File

In general, sites release the largest, oldest files first, leaving the smallest, newest files on disk. While there is only one size for a file, there are several possible ages: the age since it was last accessed, the age since it was last modified, and the age it has been resident in disk cache. In some cases, you may want to release recently accessed files before recently modified files. In other cases, a simple age derived from the most recently accessed time, modified time, and residence-changed time is preferred.

Specifying the priority for the age of a file uses one of two different sets of parameters, as follows:

Parameter Set 1: (weight_age). The weight is multiplied by the most recent access age, modify age, and residence age (in minutes) of the file. It forms the age-related portion of the file's priority.

Parameter Set 2: (weight_age_access, weight_age_modify, weight_age_residence). The age (in minutes) of each file is multiplied by the corresponding factor. The age portion of the priority is the sum of these three multiplications.

You can specify the weight_age, or any combination of weight_age_access, weight_age_modify, or weight_age_residence. However, you cannot mix weight_age with any combination of weight_age_access, weight_age_modify, or weight_age_residence.

CAUTION

You can specify weights for a file system from only one of the parameter sets. Mixing weights from both parameter sets results in an error.

The following methods illustrate how the release priority depends on the age of a file:

Method 1:	Smaller of access-age, modify-age, or residence-change-age * weight_age
Method 2:	access-age * weight_age_access + modify-age * weight_age_modify + residence-change-age * weight_age_residence

Size of File

In all cases, the size of the file (in 4-kilobyte blocks) is multiplied by a factor called weight_size to obtain the size-related portion of the priority.

File System Specific Directives

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. 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.

How to Configure the Releaser for Files in Disk Cache

It is necessary to decide the characteristics of files in cache for your site. It is wasteful to load a tape if you are only staging 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 example demonstrates, most situations are not this straightforward.

Example 1: 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 may 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.

The following example presents a better method to specify how to release the largest files first.

Example 2: 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 may want to run the releaser manually. For this, you need to know the mount point of the file system and the low water mark the releaser should attempt to reach.

Example. Assume you want to release files in the /sam1 file system until it reaches 47% full. Log in as root and type the following:

server# /opt/LSCsamfs/sbin/sam-releaser x /sam1 47 1.0

The **x** indicates compatibility with previous versions of the ASM or ASM-QFS file system and must be entered as shown. The final argument, weight-size, is overridden by a weight_size command in the command 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:

- There is no archive copy.
- The archiver set the archnodrop flag in the inode to request 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.

Chapter 7 - The Recycler

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 an historical record of file changes. Users are unaware of the recycling process as it relates to their data files.

The following topics are presented:

- Recycler overview
- Theory of operations
- How to configure the recycler
- Troubleshooting the recycler

Recycler Overview

The recycler is responsible for keeping the amount of space consumed by expired archive copies to a minimum. At any time, the capacity of a given cartridge is divided into some mixture of these three classifications of space use:

- Free space is space that is not being used by archive images.
- Current data is space being used for archive images that are current.
- Expired data is space used by archive images that are no longer current.

A newly labeled cartridge starts out with all its capacity as free space. As data is archived to the cartridge, 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 by recycling 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 only can be appended to. They cannot be rewritten in place. The only way to recycle 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 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*.

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 can be triggered by automated library utilization, by archive set utilization, or by entering the sam-recycler(1M) command.

Theory of Operations

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.

Each time it is run, the recycler performs the steps as described in the following subsections. The following subsections assume a knowledge of rearchiving. More information on rearchiving, see chapter 4, "The Archiver".

Step 1. Verify a Single Instance of the Recycler Running

The recycler maintains file /etc/opt/LSCsamfs/recycler.pid, which, through a file-level lock, prevents multiple copies of the recycler from running.

Step 2. Read the Archiver archset File

Whenever the archiver.cmd file is modified, the archiver writes a new binary representation of its commands. The recycler reads this file to obtain information such as the recycling parameters for each archive set and the information contained in the section on VSN associations.

An archive set is considered for recycling by archive set only if it is mentioned in the additional directives section of the archiver.cmd file with one or more recycling directives (bracketed by the keywords param/endparams). These archive sets are called *recycling archive sets*. For more information, see the archiver.cmd(4) man page.

Step 3. Scan the Library Catalogs

The recycler obtains the library catalog file for each automated library configured in the system's mcf file. Note that cartridges in manually loaded drives cannot be recycled.

The library catalog contains the media type, the Volume Serial name (VSN), and the space and capacity values for each volume in the automated library.

Space and capacity are reported by the drive after each file is written to a volume. These values are based on the drive's sense of how much of the volume has physically been used, and on its knowledge of the recording density on the volume. These numbers, therefore, do not take into account the data compression that the drive may be performing.

Space is the amount of space left before a volume is full. *Capacity* is the total amount of data that fits on the cartridge. A 10-gigabyte tape with 3 gigabytes written to it has a capacity of 10 gigabytes and a space of 7 gigabytes.

Step 4. Read the recycler.cmd File

The recycler reads and checks the /etc/opt/LSCsamfs/recycler.cmd file. This file contains options for processing each automated library in the system. For a complete description of the command options and syntax, see the recycler.cmd(4) man page.

The following is an example of a recycler.cmd file:

```
logfile = /usr/tmp/recycler.log
stk30  51  60  ignore  mail  root
```

The directives include the high-water mark, the minimum VSN gain, an address to which mail concerning this automated library should be sent, and the word ignore indicating that recycling should not occur for this automated library.

The high water mark allows you to set the percentage of media usage below which recycling must not occur. This percentage is the ratio of the used space in the automated library to its total capacity. As an example, an automated library that holds 10 20-gigabyte tapes, three of them 100% full and the remaining seven each 30% 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 water mark is 51% 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:

server# **chmed +c**

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

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% current data and 5% expired data, the gain obtained by recycling the cartridge is only 5%. It may not be worth moving the other 95% to retrieve this space. Setting the min-gain to 6% or more inhibits the recycler from automatically selecting this example VSN.

Another example is a cartridge with 90% expired data, 5% current data, and 5% free space that would have a gain of 90% if recycled.

The mail option specifies that the recycler send mail when recycling occurs on a given automated 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.

The ignore keeps the recycler from operating on a new candidate within the automated library.

Step 5. Read the Archiver Reserved VSN File

The archiver keeps a reserved VSN file (ReservedVSNs) that contains a list of volumes dedicated to a specific archive set. The recycler reads this file to obtain the list so that volumes can be assigned to archive sets if you are recycling by archive set. For more information, see Step 8.

Step 6. Verify that all File Systems are Mounted

The inodes contain information about the volumes used to archive a given file. Therefore, all configured ASM and ASM-QFS file systems must be mounted to allow the .inodes file to be read. If this restriction is not enforced, the recycler might undercount the number of archive images on media and erroneously conclude that current data does not exist on a cartridge.

Note that this check is not foolproof. For example, assume that you have just completed a samfsdump(1M), sammkfs(1M), and mount(1M) on a file system with the intent of running samfsrestore(1M) on the dump. If the recycler runs at this time, the requirement that all file systems must be mounted would be met, but the recycler would not know about the archive images held in samfsdump.

CAUTION

It is vitally important that you disable the recycler whenever such situations occur.

Step 7. Scan Each File System's .inodes File, Building the VSN Table

After making sure all file systems are mounted, the recycler reads each inode in each file system. For each archive copy, the VSN on which the copy resides and the size of the archive copy is accumulated into a VSN table. When this process completes, the recycler knows the amount of good data present on each VSN.

Step 8. Assign Appropriate VSNs to the Appropriate Archive Sets

The recycler assigns VSNs to the appropriate archive set by using data from the archiver's archset and reserved VSN file, the library catalogs, and the file system .inode files.

If a VSN belongs to more than one archive set, the recycler arbitrarily assigns the VSN to only one archive set. This might occur when the VSN matches all the regular expressions for the archive sets.

VSNs that are not assigned to an archive set remain assigned to the automated library from which they came. They are recycled by that automated library.

Step 9. Sort the VSN Table

The VSN table is augmented with the space and capacity values from the library catalog. The entire VSN table is sorted to first group VSNs by automated library and archive set, and then to move the most desirable VSNs earlier in the table. VSN desirability is defined as the amount of expired space on the VSN. When ties occur, the VSN with the least good data is the most desirable. This ensures that the VSN with the most to gain for the least amount of work is at the beginning of the table.

Step 10. Stop Running Now If Any Errors Were Detected

If errors are detected up to this point, the recycler exits.

Step 11. Display the Library Catalogs

For each automated library and recycling archive set, and for the entries in the historian, a table is printed showing the associated VSNs. Note that a specific VSN might appear more than once in these tables. This allows you to see each archive set in which a VSN may be used. Later, in the VSN table, the VSN appears only in its assigned archive set.

Step 12. Select Candidates for Each Automated Library Being Recycled

For each automated library and archive set that needs recycling (because its media utilization is larger than its high water mark) and is not ignoring recycling, the recycler finds a candidate VSN to recycle. If any VSN in the automated library is already marked for recycling, that VSN is automatically the candidate again. Otherwise, the first entry in the VSN table that meets the minimum gain requirement is the candidate. The recycler marks that VSN "recycling".

The recycling flag is stored in the catalog and displayed in the samu(1M) utility and in the Graphical User Interface (GUI) tools. The flag can be set manually with the chmed +c command. Once the recycling flag is set, the archiver no longer archives new images to that VSN.

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

Step 13. Display the VSN Table

The VSN table is displayed. For more information on this, see the recycler(1M) man page.

Step 14. Mark Archive Copies on Recycling VSNs for Rearchiving

Another pass through all the file system's .inodes files is needed to find the archive copies that reside on VSNs marked for recycling. Such copies are marked to be rearchived. This is how manually selected VSNs get recycled. For manually selected VSNs, the recycler recognizes the recycling flag and sets the rearchive flag on the archive copies.

Step 15. Run the recycler.sh Script

During the same .inodes file scan that occurred in step 9, the recycler counts the number of archive images on VSNs marked as recycling. If a VSN marked as recycling has no archive copies at the end of the scan, the recycler concludes that the archiver has finished rearchiving the images on the VSN.

How To Configure the Recycler

Prior to configuring the recycler, you should note the following information:

- The recycler should not be used if you have data that was placed on the VSN using the request(1) command. The recycler does not preserve removable media files created by the request(1) command.

- 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 those 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.

- The recycler is started by running the recycler(1M) command from the command line or from a cron(1) job.

Step 1: Set up the recycler.cmd File

Write the /etc/opt/LSCsamfs/recycler.cmd file as specified on the recycler.cmd(4) man page. Determine a location for the log file. Select a percentage for each automated library's high water mark and VSN minimum-percent-gain and specify these entries in the /etc/opt/LSCsamfs/recycler.cmd file. For now, also specify ignore on each line. A typical value for a high water mark might be 85%, and for min-gain, 40%, as shown in the following example. Even if you are recycling by archive set, you still should configure each automated library in the recycler.cmd file. This ensures that VSNs that do not fall into an archive set can be recycled if needed.

NOTE

Make sure that there is a space on each side of every equal sign (=) in the recycler.cmd file. Failure to include the spaces causes a syntax error, prevents the recycler from running, and creates an error message in the log file for the ASM or ASM-QFS file system.

Step 2: Edit the archiver.cmd File (Recycle by Archive Set Only)

Edit the /etc/optLSCsamfs/archiver.cmd file and add similar information for the archive sets you wish to recycle. 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.

Step 3: Run the Recycler

Run the recycler(1M) command. The recycler reads the recycler.cmd file. Examine the standard output, log, sam-log and /var/adm/messages for any error messages from the recycler. A sample recycler log follows:

```
========== Recycler Wed Jul  2 14:30:06 yyyy ===========

1 catalog:

0  Family: hp30              Path: /etc/opt/LSCsamfs/hp30_cat

  Vendor: HP                 Product: C1107A

  slot          ty   capacity      space vsn

    0           mo   652328960    648788992 OPT000

    1           mo   652328960    651802624 OPT001

    2           mo   652328960    651802624 OPT002

    3           mo   652328960    630462464 OPT003

    4           mo   652328960    497533952 OPT004

    5           mo   652328960    567013376 OPT005

    6           mo   652328960    651802624 OPT006

    7           mo   652328960    651802624 OPT007

    8           mo   652328960    548516864 OPT008

    9           mo   652328960    651802624 OPT009

   10           mo   652328960    434416640 OPT010

   11           mo   652328960    486307840 OPT011

   12           mo   652328960    587621376 OPT012

   13           mo   652328960    587872256 OPT013

   14           mo   652328960    559387648 OPT014

   15           mo   652328960    632983552 OPT015

   16           mo   652328960    633071616 OPT016
```

```
   17               mo    652328960    587646976 OPT017

  Total Capacity:  11741921280 bytes, Total Space Available: 10660636672
bytes

  Robot media utilization 9%, high 10% VSN_min 10%


18 VSNs:

      ------In use-------   -------Percent-------

Recycle  Ar sections   Bytes   In use  Obsolete Free   Library:Type:VSN

  n        0           0     0      33     67    hp30:mo:OPT010

  n        0           0     0      25     75    hp30:mo:OPT011

  n        0           0     0      23     77    hp30:mo:OPT004

  n        0           0     0      15     85    hp30:mo:OPT008

  n        0           0     0      14     86    hp30:mo:OPT014

  n        0           0     0      13     87    hp30:mo:OPT005

  n        0           0     0       9     91    hp30:mo:OPT013

  n        0           0     0       9     91    hp30:mo:OPT012

  n        0           0     0       3     97    hp30:mo:OPT003

  n      712     47741263     7       2     91    hp30:mo:OPT017

  n    13188      2373604     0       2     98    hp30:mo:OPT016

  n    13084      2328244     0       2     98    hp30:mo:OPT015

  n      324      1327104     0       0    100    hp30:mo:OPT000

  n        0           0     0       0    100    hp30:mo:OPT002

  n        0           0     0       0    100    hp30:mo:OPT009

  n        0           0     0       0    100    hp30:mo:OPT001

  n        0           0     0       0    100    hp30:mo:OPT006

  n        0           0     0       0    100    hp30:mo:OPT007
```

Recycler finished.

Step 4: Create a crontab Entry for the Recycler

If the system is performing as expected, you are ready to make a crontab entry for the super user to run the recycler(1M) periodically. You may 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 recycler periodically:

```
0,2,4,6,8,10,12,14,16,18,20,22 * * * *  /opt/LSCsamfs/sbin/recycler
```

Step 5: Remove ignore Directives

Remove the -recycle_ignore and ignore directives from the archiver.cmd and recycler.cmd configuration files. You are now recycling.

Step 6: Create a recycler.sh File

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/LSCsamfs/examples/recycler.sh shows how to relabel a recycled VSN and send mail to the super user.

```
#  !/bin/csh -f

#

#  /opt/LSCsamfs/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 - MID in the library where the VSN is located
```

```
#
# $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 modifier, 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/LSCsamfs/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/LSCsamfs/sbin/chmed -R $5.$2
# /opt/LSCsamfs/sbin/chmed -W $5.$2
# if ( $5 != "d2" ) then
# if ( $1 != "od" ) then
# /opt/LSCsamfs/sbin/${1}label -w -vsn $2 -old $2 $4\:$3
# if ( $status != 0 ) then
# set stat = 1
# endif
# else
# /opt/LSCsamfs/sbin/${1}label -w -vsn $2 -old $2 $4\:$3\:$7
# if ( $status != 0 ) then
# set stat = 1
# endif
# endif
# else
# /opt/LSCsamfs/sbin/${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
#I  mages.  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 ASM samu or
#  robottool programs to export this medium from the historian catalog to
#  suppress this message.
#  /eof
#  endif
#  echo `date` $* done >>  /var/opt/LSCsamfs/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/LSCsamfs/recycler.sh.log
```

```
#  exit 0


#  The default action is to mail a message reminding you to set up this

#  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/LSCsamfs/sbin/recycler.sh script was called by the ASM recycler

with the following arguments:

Media type: $5($1)  VSN: $2  Slot: $3  Eq: $4

Library: $6

/opt/LSCsamfs/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/LSCsamfs/recycler.sh.log

exit 0
```

The recycler called the /opt/LSCsamfs/sbin/recycler.sh script with the following arguments:

Media type: $1 VSN: $2 Slot: $3 Eq: $4

The /opt/LSCsamfs/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 disposition of 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) man page.

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 the 123 inodes that appear to be valid with archive copies on the volume. Because the inodes are not part of the directory tree, they are not accessible by a file name. The archive copies are not active.

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, message 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 only occur as the result of a system crash that partially corrupted the .inodes file

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

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

Chapter 8 - Graphical User Interface (GUI) Tools

This chapter describes the GUI tools used to manage the devices in an ASM or ASM-QFS environment. There are two GUIs used to manage robots, devices, and media mount requests:

- libmgr(1M) - This GUI is based on Java technology. It provides a single interface to all automated libraries and devices and can be customized for operations at your site.

- samtool(1M) – This GUI is based on X Window System technology. It consists of three interfaces: robottool, devicetool, and previewtool.

To use the tools, you need to be familiar with using a window system based on X Window System technology, such as the Common Desktop Environment (CDE) or the Open Windows environment. You also need to be familiar with the mouse buttons.

This chapter contains the following topics:

- Operator privilege levels
- Using libmgr(1M)
- Using samtool(1M)
- Using robottool(1M)
- Using devicetool(1M)
- Using previewtool(1M)

NOTE

In addition to these GUIs, you can manage devices in an ASM or ASM-QFS environment using the curses-based operator tool, samu(1M). For instructions on using the samu(1M) utility, see chapter 9, “The samu(1M) Operator Utility”.

Operator Privilege Levels

The GUI tools described in this chapter are, by default, for superusers only. As a site administrator, you can define operational authority that is not superuser privileged, but has 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/LSCsamfs/defaults.conf file. Users with root authority have full access to functions within samtool. Users who are part of the operator group have access removed for 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 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, import and export cartridges, and respond to cartridge load requests.

Before using libmgr(1M), you must have the optional Java runtime environment and ASM GUI packages installed on your system. To check for these packages, enter the following commands:

server# **pkginfo LSCjre**

server# **pkginfo LSCgui**

For information on installing the LSCjre and LSCgui packages, see the *ASM Installation and Configuration Guide*.

How to Start the Library Manager

To start the library manager, enter the following command at the operating system prompt:

server# **libmgr&**

How 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/LSCsamfs/SamGUI.rsc resources file. Without any changes, libmgr displays device titles and images based upon the device's product ID, vendor ID, and equipment number as defined in the mcf file.

The SamGUI.rsc file allows you 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.

The Library Manager Display

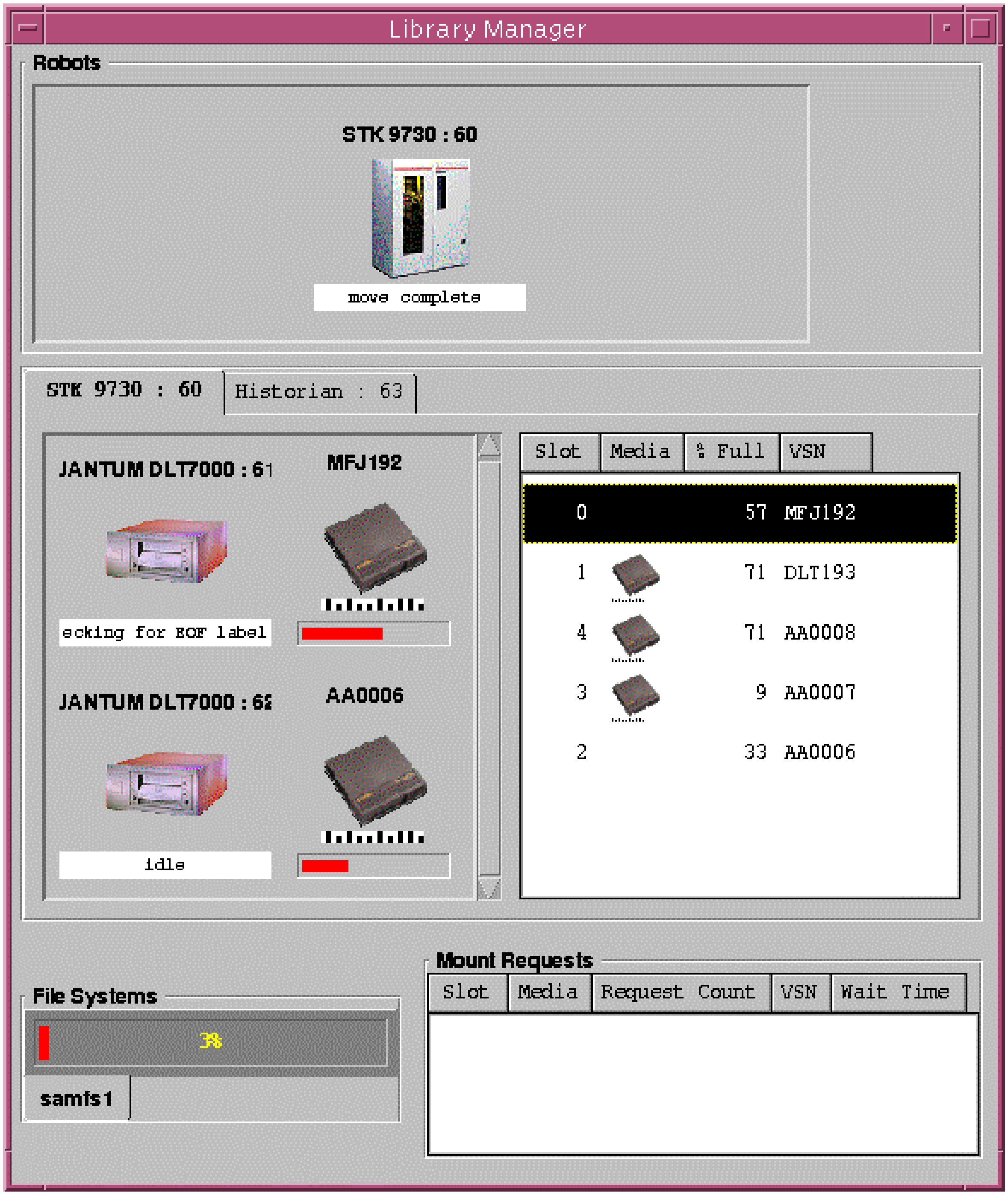
An example of the library manager display is shown in this subsection. The display consists of objects that can be manipulated by a mouse. Most objects respond to the mouse as follows:

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.

Three panels are shown in the example:

- The libraries panel on top (this doesn't appear if you don't have any robots configured)

- The catalog panel in the middle
- The file systems and mount request panel on the bottom



Robot Operations

Place the pointer on the desired robot image, then:

Desired action	Mouse button	Menu pick
Turn automated library to on, off, or down	Right click	Choose On, Off, or Down.
Import media	Right click	Choose Import.
Unload VSNs from robot catalog	Right click	Choose Unload. The robot's catalog is emptied and the robot set to off. Set robot to on to reset.
Fully audit robot	Right click	Choose Audit.

Media Operations

In the catalog panel, select the desired robot. Select the medium with a left click, then:

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.

Media Drive Operations

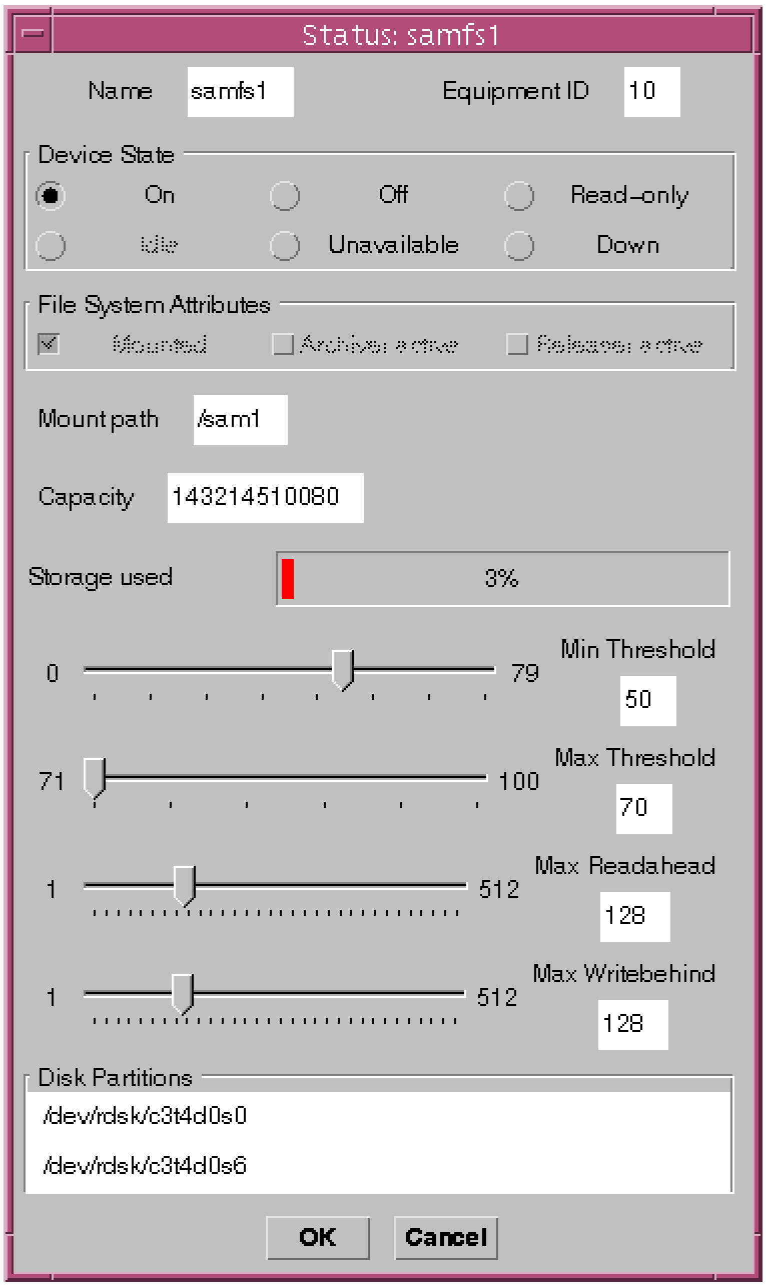
Place the pointer on the desired media drive image, then:

Desired action	Mouse button	Menu pick
Label or relabel media.	Right click	Choose Label. Enter VSN, blocksize, and optionally relabel or erase.

Desired action	Mouse button	Menu pick
Turn drive to on, off, unavailable, or down.	Right click	Choose On, Off, Unavailable, or Off.

File System States and Attributes

To view the file system states and attributes and make changes, select the desired file system tab, then double click on the bar graph. A detailed information window for this file system appears. An example follows.



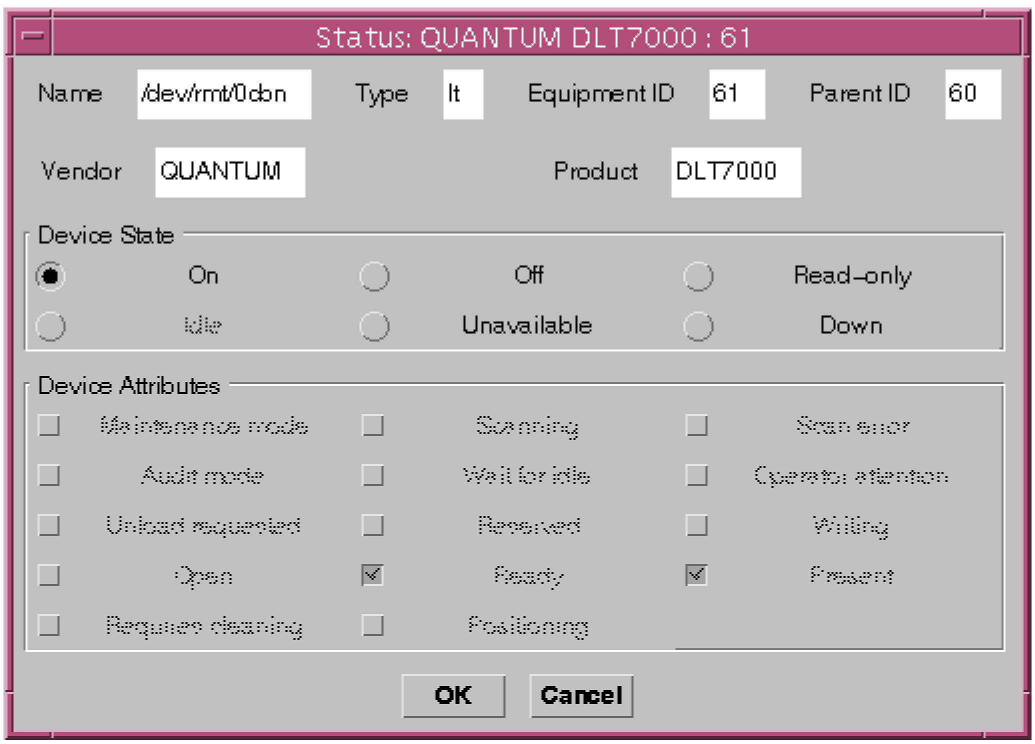
The file system device states and device attributes are described on the libmgr(1M) man page under the Icon Attributes heading. Any changes to the attributes or states are enabled by selecting the button and clicking OK.

NOTE

Any changes to the minimum release threshold, maximum release threshold, and maximum contiguous block settings can only be set for the current ASM or ASM-QFS session. When the sam-initd daemon is stopped, the system resets to the default settings or to the settings for this file system as listed in the /etc/vfstab file.

Media Drive States and Attributes

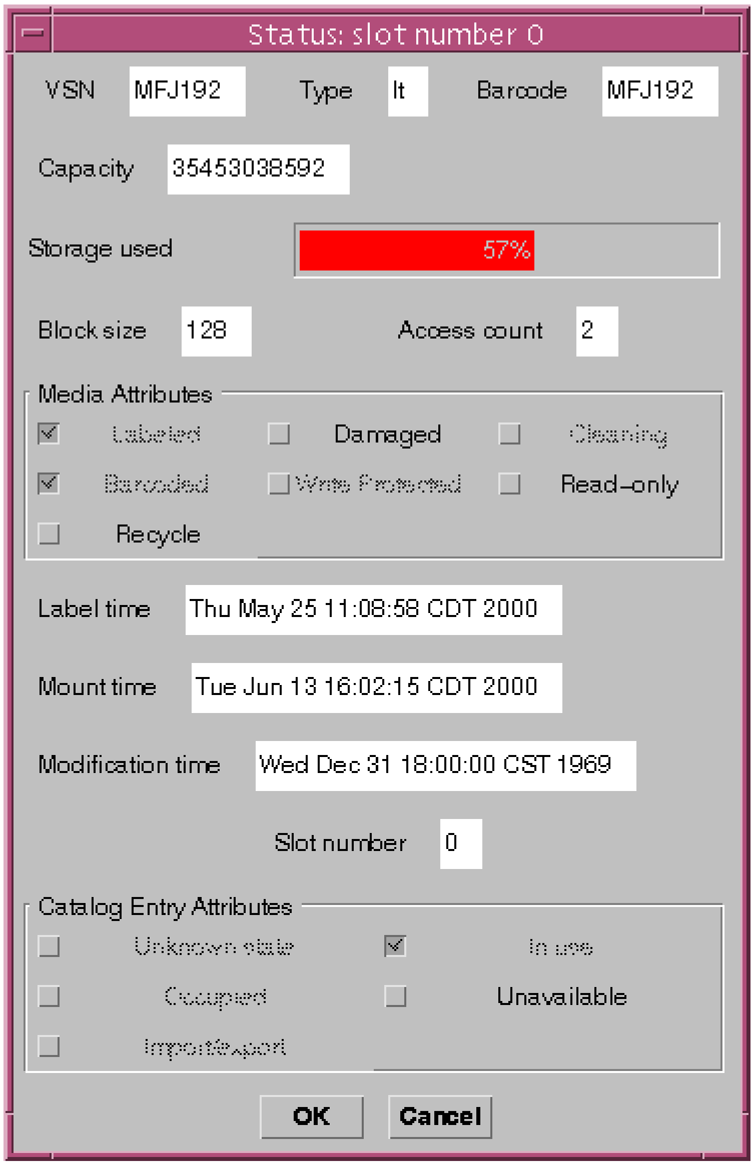
To view the media states and attributes and make changes, double click on the desired media drive image. The device state and attributes window for this device appears. An example follows:



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.

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. An example VSN catalog display window follows:



Using samtool(1M)

The following subsections describe the following topics:

- How to start and quit samtool(1M)

- Selecting a tool
- Updating displays
- Screen resources
- Getting online help

How to Start and Quit samtool

To start samtool, enter the following command at the operating system prompt:

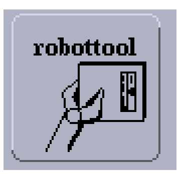
server# **samtool&**

The system displays the samtool group, as follows:

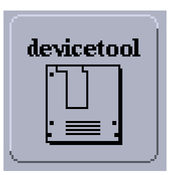
To exit samtool, right click on the top bar of the window and select QUIT or Close.

Selecting a Tool

samtool displays an icon for each of its tools. To invoke a tool, left click on the icon of the tool you want to use.



Allows you to view and manage information on robots configured within the ASM or ASM-QFS environment.



Allows you to view and manage information on devices configured within the ASM or ASM-QFS environment.

Allows 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, enter robottool on the command line. To start a tool in background, type an ampersand (&) after the tool name. For example, to start robottool in background, enter robottool&.

Updating Displays

By default, all samtool 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 when needed. The update button, the refresh checkbox, and the refresh field in the upper right of each display control updates.

To change the refresh rate:

1) Make sure automatic refresh is enabled. That is, make sure the refresh checkbox contains a check to indicate it is enabled.

2) Type in a new refresh rate in the refresh field or use the increase/decrease setting buttons.

To immediately update a tool display, click the Update button located in the upper right of the window.

To enable or disable automatic refresh, click on the refresh. Automatic refresh is enabled when the refresh checkbox contains a check.

Screen Resources

The font for panel lists used in samtool displays can be changed 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

Getting Online Help

To display online help for samtool:

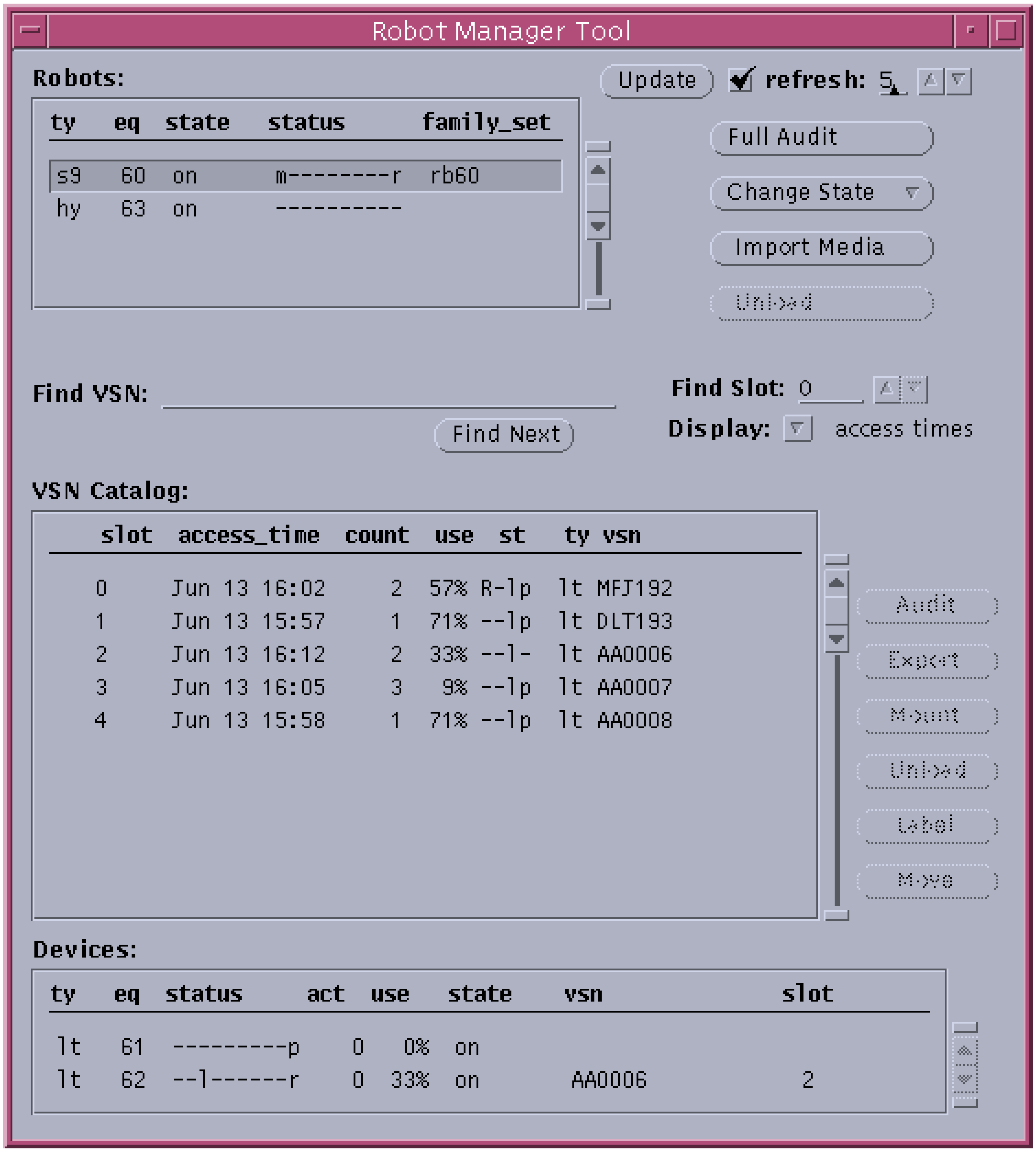
- Left click on the Help button to display general help on samtool and its operation.

- 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.

The following figure shows a sample robottool display:



The robottool display contains three areas:

- Robots
- VSN Catalog
- Devices

The following subsections describe the content of these three display areas.

How to Start robottool

To start samtool, enter the following command at the operating system prompt:

server# **robottool&**

Robots

The Robots area lists all robots configured within an ASM or ASM-QFS environment. The following information is displayed for each robot:

Information	Description
ty	Equipment type.
eq	Equipment ordinal.
state	State of equipment. See “Changing the State of a Robot”.
status	Status of the robot. See “Viewing Status Information”.
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. The following information is available for each VSN:

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. See “Viewing VSN Status Information”.

Information	Description
ty	Media type.
vsn	Volume serial name.

The VSN display includes either access time or barcode information for all VSNs. To select one or the other, see “Tailoring the VSN Catalog Display” later in this chapter. 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. The following information is displayed:

Information	Description
ty	Equipment type.
eq	Equipment ordinal.
status	Status of the device. See “Viewing Status Information”.
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, use devicetool, described later in this chapter.

Viewing Status Information

The following table describes the status strings:

Status bit	Meaning for device	Meaning for file system
s---------	Media is being scanned.	
m---------		The file system is currently mounted.

Status bit	Meaning for device	Meaning for file system
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.	
--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----	The device is reserved.	
------w---	A process is writing on the media.	
-------o--	The 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.	

Status bit	Meaning for device	Meaning for file system
---------r	Device is spun up and ready.	The file system's disk space is being released.
---------p	Device is present.	

Managing Robots

This subsection describes the actions you can perform on a selected robot. The possible actions are as follows:

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.

Performing a Full Audit

To perform a full audit of all volumes in a robot:

1) Select the robot in the list of available robots.

2) Left click on the Full Audit button. You are prompted to confirm the operation.

   The system performs a full audit of every volume in the robot.

Changing the State of a 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. Possible states include:

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 to the mailbox. If 30 seconds pass without inserting cartridges, 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 provides 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 subsection gives instructions for working with volumes.

- Tailoring the VSN catalog display
- Selecting a VSN
- Finding a VSN
- Auditing a VSN
- Exporting a volume
- Mounting and unloading a volume
- Labeling media
- Moving media
- Viewing VSN status information

<u>Tailoring the VSN Catalog Display</u>

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:

  Right click on the Display button shown below and select barcodes:

The system displays barcodes rather than access times.

- To display access times:

  Right click on the Display button and select access times.

  The system displays access times rather than barcodes.

Selecting a VSN

- To select a VSN:

  Left click on the VSN you want to select.

Finding a VSN

- To search for and select a volume by VSN:

  Type a VSN name or beginning pattern to match in the Find VSN field shown below:

  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, press 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 shown below:

  You can also use the up/down buttons to increase or decrease the slot number. If no VSN is found, an error message results.

  To sequence through to the next slot number, press the Find Next button.

Auditing a VSN

To perform an audit on a selected VSN:

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.

Exporting a Volume

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.

Mounting and Unloading a Volume

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.

Labeling 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 where data begins to be written on the cartridge.

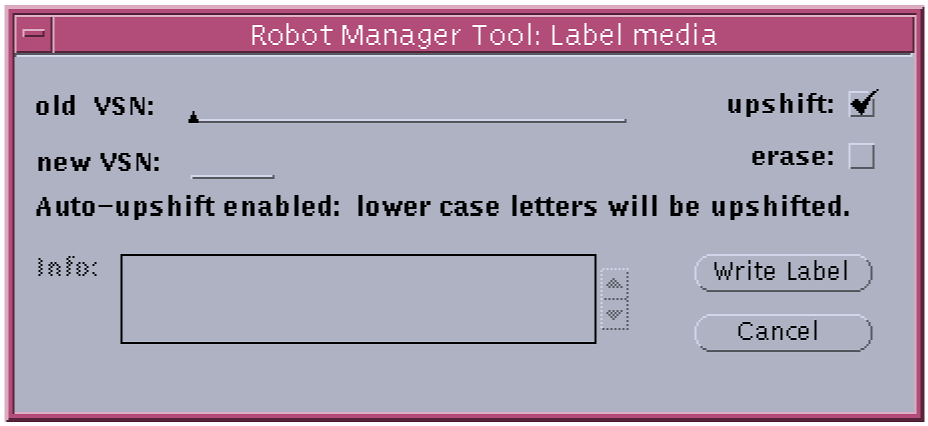
CAUTION

Labeling a volume causes the loss of previously written data on that volume. Be sure that this is your intention prior to proceeding.

To label a volume:

1) Select the VSN you want to label.

2) Left click on the Label button. The following dialog box is displayed:



3) Enter 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. Note that erasing media can require a significant amount of time. Note that data is always lost during a label. Erasing additionally overwrites each sector on the volume.

5) Click the Write Label button.

6) If an error is detected, an error checkbox and message appears in the Label media window above the info box. To acknowledge the error, click in the checkbox, and the error message is removed.

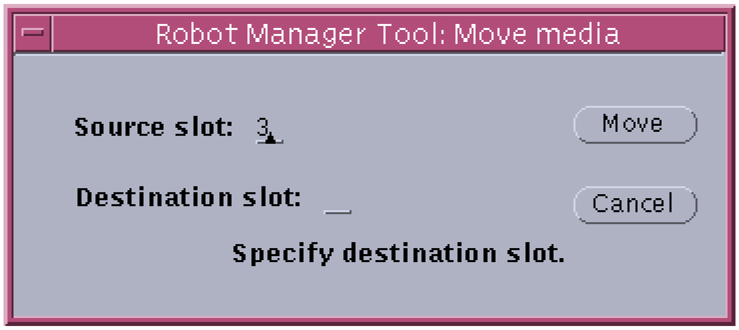
Possible errors include an invalid VSN or an old VSN that does not match the VSN of the volume in the selected slot.

Moving Media

To move a volume to another slot:

1) Select the volume you want to move.

2) Left click on the Move button. The following dialog is displayed:



3) Enter the following in this box:

- For Source slot, you can enter a new slot number by either double-clicking on 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 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. The following table describes the possible status flags:

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 non-robotic devices, use devicetool. The information displayed is the same as that displayed in the devicetool medium-specific displays.

Using devicetool(1M)

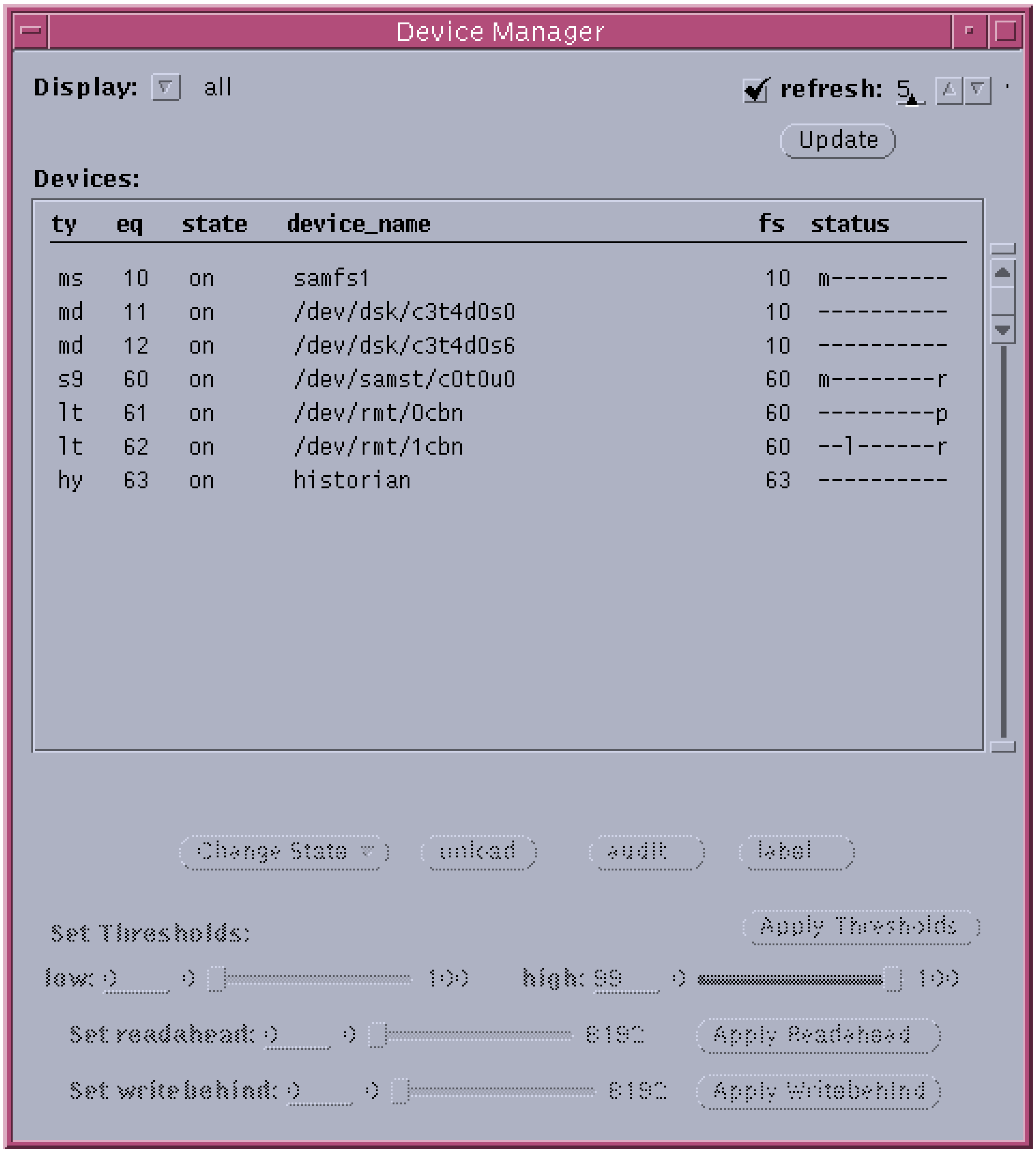
devicetool(1M) presents a GUI tool for viewing information about and managing devices associated with ASM and ASM-QFS.

How to Start devicetool(1M)

To start devicetool, enter the following command at the operating system prompt:

server# **devicetool&**

The following figure shows a sample devicetool display:



Changing the Display Format

devicetool 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.

To change the format of the display:

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 on the Media button. A pull-down menu is displayed.

4) Select a media type for the display.

Viewing the Display Fields

Some display fields are common to all displays. Other fields are displayed only for certain display formats. The following table lists all the possible fields displayed in alphabetical order:

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: 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 not available for access. down - the device is available only for maintenance access.

Field	Description
status	Device status.
ty	Device type.
used	Percentage of space used.
vsn	Volume serial name assigned to the volume or "nolabel" if the volume is not labeled. This field is blank when no medium is present in the transport, or the device is off.
wb	Maximum writebehind on this file system in units of 1-kilobyte blocks.

Controlling Devices

This subsection gives instructions for controlling devices:

- Changing the device state
- Unloading a device
- Auditing a volume in a device
- Labeling a volume in a device

A device is selected from the display by clicking 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. Possible actions are change state, unload, audit, and label.

The device state can be changed with the Change State button. 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 may select a device state. Possible states are on, idle, off, and down.

Changing the Device State

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.
   - Possible states include:

Current State	Possible Next State
ON	IDLE, OFF
IDLE	Automatically goes to OFF when IDLE
OFF	DOWN, ON
DOWN	OFF

Unloading a Device

To unload a volume:

1) Select the device you want to unload.
2) Left click on the Unload button.

   The robot unloads the selected device.

Auditing a Device

To perform an audit on a selected 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.

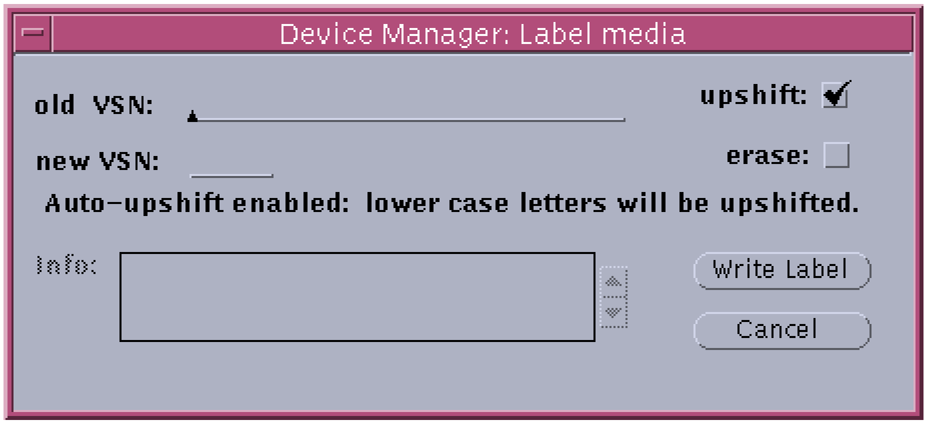
NOTE

To perform an audit for every VSN in a robot, select the robot in the Robot Display and click the Full Audit button.

Labeling Media

To label the volume in a device:

1) Select the device for which you want to label media.
2) Left click on the Label button. The following dialog is displayed:



3) Enter 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. Note that erasing media can require a significant amount of time.

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 media in the selected slot.

CAUTION

Labeling a volume causes the loss of all data on that volume.

Setting Thresholds

For disk sets, you can set low and high thresholds for the disk set:

To set thresholds:

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.

Setting Readahead and Writebehind

You can set the maximum number of contiguous 1-kilobyte blocks for readahead and writebehind on a disk set. To set readahead or writebehind:

1) Select the disk set for which you want to set readahead or writebehind.

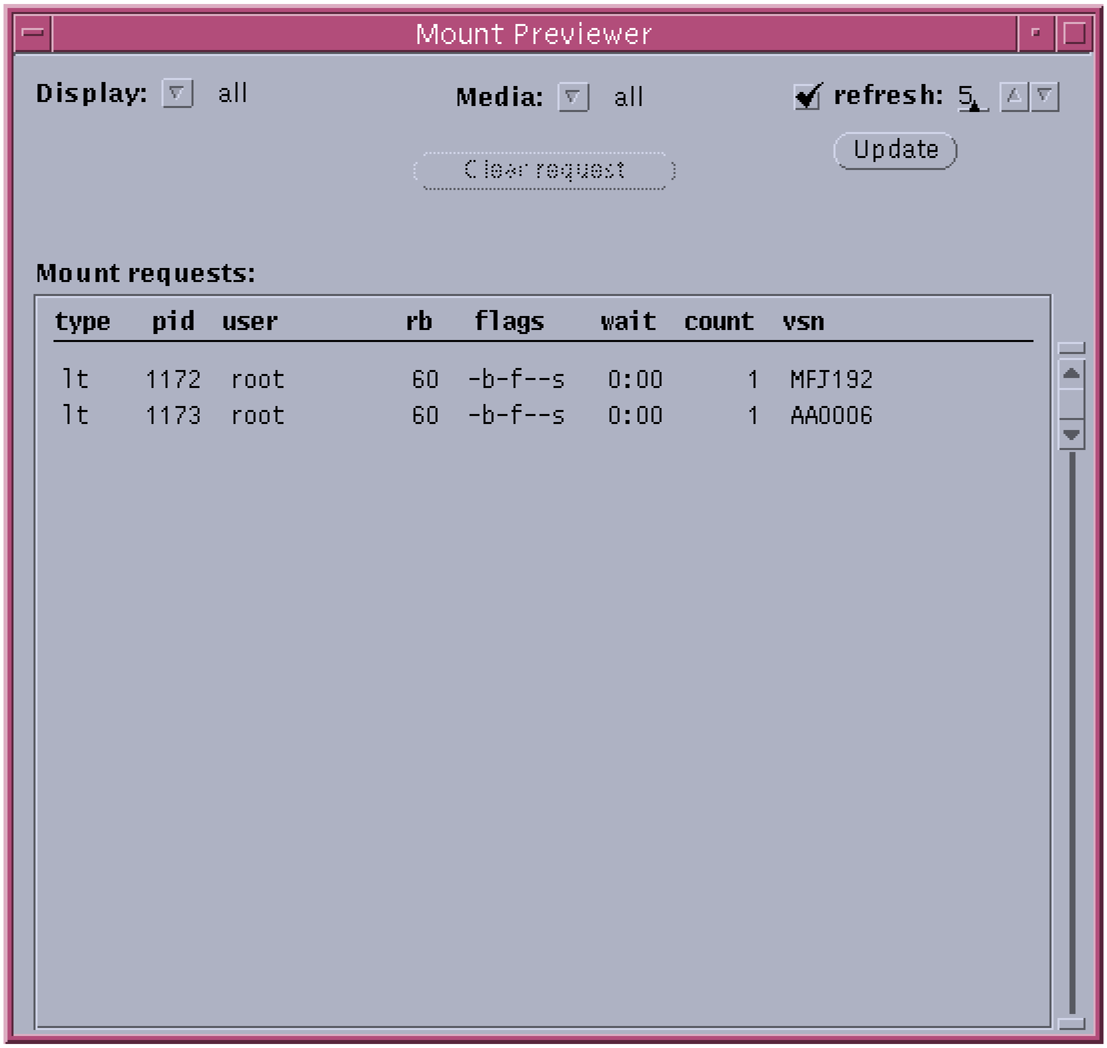
2) Type a number that specifies the number of contiguous 1-kilobyte blocks for either reading ahead or writing behind, whichever one you are setting. Alternatively, you can use the appropriate slider bar to increase or decrease the number.

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)

previewtool 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.

The following figure shows a sample previewtool display:



Changing the Display Format

To change the display format:

1) Right click on the Display button. Select one of the options displayed.

2) Select a display type.

- If you select 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.

Viewing Displayed Fields

The following table describes the fields displayed by previewtool.

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 the description after this table.
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 name assigned to the media.

The following flags are displayed in the flags field:

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

Clearing Mount Requests

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.

Chapter 9 - The samu(1M) Operator Utility

This chapter provides instructions for controlling the devices configured within your ASM and ASM-QFS environment through the samu(1M) operator utility.

The following topics are presented:

- Overview
- Operator commands
- Operator displays

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 allow you to monitor ASM and ASM-QFS devices and file system activity.
- Commands that allow 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 may be different depending on your terminal model and the devices configured in your 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 the samcmd(1M), see the samcmd(1M) man page.

The following subsections describe how to start and stop the samu(1M), interact with the utility, access the help windows, and view operator displays.

Invoking samu(1M)

To start samu(1M), enter the samu command from the UNIX command line as follows:

server# **samu**

The system starts samu(1M) and shows the help display. Press CTRL-f to move to the next help screen, which shows the keys that control the displays.

The samu 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).

Stopping samu(1M)

To exit samu(1M), enter one of the following:

- Press the q key
- Enter :quit
- 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, the keys described in the following table can be used to control the display. The exact function of these keys depends upon the display being viewed at the time. For information on display-specific key operations, see the samu(1M) man page. The descriptions shown in the following table are the typical responses:

Key	Function	Display

Key	Function	Display
CTRL-b	Previous file system	:a, a
	Page backward	c, h, o, p, s, t, u, v, A, J, M
	Previous inode	I
	Previous sector	S
	Previous equipment	T, U
CTRL-d	Half-page forward	c, p, s, u, A, J, M
	Next robot catalog	v
	Page forward	h, S
	Page arcopies forward	a
CTRL-f	Next file system	:a, a
	Page forward	c, h, o, p, s, t, u, v, A, J, M
	Next inode	I
	Next sector	S
	Next equipment	T, U
CTRL-k	Advance display format	A, I, S
	Select (manual, robotic, both, priority)	p
	Advance sort key	v
	Toggle path display	n, u
CTRL-u	Half-page backward	c, p, s, u, A, J, M
	Previous robot catalog	v

Key	Function	Display
	Page backward	h, S
	Page arcopies backward	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 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 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. 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

Viewing Operator Displays and Status Codes

The following subsections describe viewing operator displays and status codes.

Operator Display Keys

Operator displays are viewed by simply pressing the key corresponding to each display. The lowercase keys a through v display operational information.

If samu 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 previously described.

Key	Description
a	Displays archiver status information.
c	Displays configuration information, including both file system and device definitions.
h	Displays the last help window viewed.
l	Displays ASM license information.
m	Displays status information for mass-storage file systems.
n	Displays removable media I/O activity.
o	Displays status information for optical disks.

Key	Description
p	Displays removable media mount requests. • 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. Mount requests are displayed in three formats: both manual and robotic requests, manual requests only, or robotics requests only. Enter CTRL-k to advance the formats.
r	Displays status information for removable devices such as optical disks and tapes. • Enter only r to display status information for all removable devices currently selected. • Enter :r *media_type* to display status information for devices of a given removable media type.
s	Displays a summary of status for all devices.
t	Displays status for all tape devices.
u	Displays all files in the staging queue.
v	Displays a VSN catalog for a robot.

Status Codes

The m, o, r, s, and t operator displays show status codes. Status codes are displayed in a 10-position format, reading from left (position 1) to right (position 10).

The status codes in this subsection do not apply to the samu(1M) v display.

The following table defines the valid status codes for each position.

Status bit	Meaning for device	Meaning for file system
s---------	Media is being scanned	
m---------		File system is currently mounted
M---------	Maintenance mode	

Status bit	Meaning for device	Meaning for file system
-E--------	Device received an unrecoverable error in scanning	
-a--------	Device is in audit mode	File system is being archived
--l-------	Media has a label	
---I------	Waiting for device to idle	
---A------	Needs operator attention	
----C-----	Needs cleaning	
----U-----	Unload has been requested	
-----R----	The device is reserved	
------w---	A process is writing on the media	
-------o--	The device is open	
--------P-	Device is positioning (tape only)	
--------F-	For robots, all storage slots occupied. For drives, media is full.	
---------R	Device is ready and the media is read-only	
---------r	Device is spun up and ready	File system’s disk space is being released
---------p	Device is present	
---------W	Device is write protected	

Device State Codes

The c, m, o, r, s, and t operator displays show device state codes. These codes represent the current access state for the device. The following table defines the valid state codes:

Device State	Description
on	The device is available for access. For certain displays, this state may be superseded by the states ready/notrdy.
ro	The device is available for read-only access. Like on, this state may be superseded for certain displays by ready/notrdy.
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 as 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.

Device State	Description
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 subsection describes the following types of operator commands:

- Display control commands
- Device commands
- Robot commands
- Archiver commands
- Miscellaneous commands

NOTE

Operator commands must be entered from samu(1M). These commands are invalid if you enter them from the UNIX command line.

All samu(1M) commands are prefaced with a colon (:) when entering to designate that a command line command is being entered and not a series of hot keys.

Display Control Commands

The display control commands are as follows:

:a [filesystem]

The a command displays the archiver status.

Without the *filesystem* argument, 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.

With the *filesystem* argument, the archiver status display shows the number of regular files, offline files, archived files, archive copies and directories, plus the file systems, mount points, inode activity, and interval.

:p [media]

The p command selects the media type for the mount requests display. For *media*, specify either the media type, as defined on the media(5) man page, or the keyword all. If all is specified, the operator utility displays pending mount requests for all removable media devices. If *media* is not specified, the utility displays mount requests for all currently selected devices.

:n [media]

The n command selects the media type for the removable media I/O activity display. For *media*, specify either the media type, as defined on the media(5) man page, or the keyword all. If all is specified, the operator utility displays activity for all removable media devices. If *media* is not specified, the utility displays activity for all currently selected devices.

:r [media]

The r command selects the device type for the removable media status display. For *media*, specify either the media type, as defined on the media(5) man page, or the keyword all. If all is specified, the operator utility displays the status for all removable media devices. If *media* is not specified, the utility displays the status for all currently selected removable media devices.

:v [eq]

The v command selects the library VSN catalog for display. *eq* is the equipment ordinal of the library where the catalog resides. You can view VSNs of the historian's catalog by choosing its equipment ordinal or by invoking this command as follows:

:v historian

If *eq* is not specified, the utility displays the catalog for the currently selected library.

Device Commands

The device control commands are as follows:

:devlog eq [event ...]

The devlog command sets device logging options. *eq* is the equipment number of the device from the mcf file and *event* is one or more event types from the following list:

all	none	default

detail	err	label
msg	mig	retry
stage	syserr	time
module	event	date

:down eq

The down command terminates operation on the specified device *eq*, where *eq* is the equipment ordinal.

:idle eq

The idle command restricts access to specified device *eq* by preventing new connections to the device. Existing operations continue until completion. *eq* is the equipment ordinal.

:readahead eq contig

The readahead command specifies the maximum number of bytes that can be read ahead by the file system. *eq* is the equipment ordinal for the file system. *contig* is in units of 1-kilobyte blocks and must be an integer such that 1 < *contig* < 8192. *contig* 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 equipment number 3:

:readahead 3 256

This value can also be configured in the samfs.cmd file by specifying the readahead directive. For more information, see the samfs.cmd(4) man page.

:off eq

The off command logically turns off the specified device. *eq* is the equipment ordinal.

:on eq

The on command logically turns on the specified device *eq*, where *eq* is the equipment ordinal.

:partial eq size

The partial command sets the number of kilobytes to leave online after release of the file. *eq* is the equipment ordinal for the file system, and *size* is the number of kilobytes to leave online. The default *size* is 16.

:thresh eq high low

The thresh command sets the high and low thresholds for the specified file system equipment to control file archiving. *eq* is the equipment ordinal of the storage family set, *high* is the high threshold, and *low* is the low threshold. For example, the following command sets the high threshold to 50% and the low threshold to 40% for the storage family set whose file system equipment ordinal is 10:

:thresh 10 50 40

For more information on thresholds, see chapter 1, “Overview”, chapter 6, “The Releaser”, chapter 8, “Graphical User Interface (GUI) Tools”, and chapter 11, “Advanced Topics”.

:unload eq

The unload command unloads the mounted media for the specified removable media device *eq*, where *eq* is the equipment ordinal. For magazine devices, the unload command unloads the mounted cartridge and ejects the magazine.

:unavail eq

The unavail command selects the specified equipment with equipment ordinal *eq* and makes it unavailable for use with the ASM for ASM-QFS file system.

:writebehind eq contig

The writebehind command specifies the maximum number of bytes that can be written behind by the file system. *eq* is the equipment ordinal for the file system. *contig* is in units of 1-kilobyte blocks and 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 equipment number 50:

:writebehind 50 256

This value can also be configured in the samfs.cmd file by specifying the writebehind directive. For more information, see the samfs.cmd(4) man page.

Robot Commands

The robot control commands are as follows:

:audit eq

The audit command causes the specified robotic device to mount each volume, read the VSN, and rebuild the library catalog. *eq* is the equipment ordinal.

:export eq:slot
:export mt.vsn

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. *eq* is the equipment ordinal or device name. *slot* is the decimal 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. *mt* is the media type; for information on valid media types, see the media(5) man page. vsn is the volume to mount.

:import eq

The import command causes the specified robotic device to allow you to add a cartridge. *eq* is the equipment ordinal.

:load eq:slot
:load mt.vsn

The load command allows you to load by either 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. *eq* is the equipment ordinal or device name. *slot* is the decimal slot number containing the volume you want to load.
- If loading by logical identifier, the specified robotic device to load mounts a labeled volume in to a drive. *mt* is the media type; for information on valid media types, see the media(5) man page. vsn is the volume to mount.

Archiver Commands

The archiver commands are as follows:

:arrun

The arrun command overrides an existing wait command in the archiver.cmd file. arrun sends a SIGUSR1 to the archiver and begins archiving as if the wait command were removed.

:arrestart

The arrestart command 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 removable media may not complete and must be repeated, wasting space on the media.

:artrace [options]

The artrace command sets archiver trace options. For a complete listing of all trace options, see the archiver.cmd(4) man page.

Miscellaneous Commands

The following are the miscellaneous samu(1M) commands:

:mount mntpt

The mount command selects an ASM file system.

:open eq

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.

:read addr

The read command reads the specified sector from the currently opened disk device. You must open the device before it can be read. *addr* is the hexadecimal sector address.

:refresh i

The refresh command sets the time interval for refreshing the display window and enables display refreshing. *i* is the time interval in seconds. The CTRL-r key sequence toggles display refreshing on and off.

:snap [filename]

The snap command sends a snapshot of a display window to *filename*, which is the name of a file to receive the display information. The default file name is snapshots.

As an aid to reporting problems, 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 ASM support personnel.

:clear vsn [index]

The clear command clears the specified VSN from the removable media mount requests display (see "(p) - Removable Media Mount Requests Display" later in this chapter). 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.

:quit
:q

The quit command (q for short) causes the samu operator utility to exit.

:! shell command

The ! command allows you to run a shell command without leaving the samu operator utility.

Operator Displays

The following subsections describe the operator displays. Examples are provided, and when necessary, displays are followed by a table describing the fields displayed.

For displays that overflow the screen area, the word more appears on the bottom of the screen indicating that the display information contains additional information.

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	hpc171	pt01		yes	5	0	On	hp70

more

(a) - Archiver Display

The archiver display shows the status of the archiver on a per file system basis. To view the archiver display, press the a key.

Sample Display

This display is obtained by pressing the a key. It shows activity and statistics for a single file system (set by entering :a *filesystem*).

```
Archiver status                samu   3.5.0    Tue Apr 25 10:58:52
License: License never expires.

samfs1 mounted at /sam1
regular files       658  981.3M
offline files         1  165.7k
archdone files      651  981.1M
copy1               649  980.9M
copy2               648  980.8M
copy3                 0
copy4                 0
directories           5   48.2k
Sleeping until Tue Apr 25 10:59:23 2000
```

Field Descriptions

Field	Description
samfsl 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.
copy3	Number of files and total size for archive copy #3.
copy4	Number of files and total size for archive copy #4.

Field	Description
Directories	Number of directories and total size.
sleeping until	Indicates when archiver runs again.

(b) - Configuration Display

The configuration display shows the connectivity of your ASM or ASM-QFS configuration. To view the configuration display, press the c key.

Sample Display

```
Device configuration:          samu   3.5.0 Tue Apr 25 12:13:49
License: License never expires.

Ty   eq    state  device_nam3          fs    family_set
Ms   10    on     samfs1               10    samfs1
Md   11    on     /dev/dsk/c0t0d0s0    10    samfs1
Xt   100   on     /dev/rmt/0cbn
Sl   200   on     /dev/samst/c1t3u0    200   sl200
At   201   on     /dev/rmt/1cbn        200   sl200
At   202   on     /dev/rmt/2cbn        200   sl200
Ss   300   off    /etc/fs/samfs/rmt    300   300 neptuners
Hy   301   on     historian            301
```

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: 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

Field	Description
	progress continue until completion.
device_ name	Path to the device (for type ms, file system name).
fs	File system equipment ordinal.
family_ set	Name of the storage family set or library to which the device belongs. Storage family set is for magnetic disks only.

(c) - License Display

The license display shows the licenses and expiration dates for each ASM product. To view the configuration display, press the l key.

Sample Display

The following 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/LSCsamfs/LICENSE.3.5

- The following information is displayed for the system:
- Expiration information
- Host ID
- ASM products and features enabled
- Equipment/media combinations

```
License Information                samu   3.5.0 Tue Apr 25 12:16:59
License: License never expires.

hostid = 7232855a
License never expires
Remote sam server feature enabled
Remote sam client feature enabled
Migration toolkit feature enabled
Fast file system feature enabled
Direct media access feature enabled
```

Segment feature enabled

Robot type Spectra Logic Library is present and licensed

30 at slots present and licensed

(d) - 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

The following display shows the status for two file system drives that make up two sets. Member drives are indented 1 space and appear directly below the file system to which they belong.

```
Mass storage status                 samu   3.5.0 Tue Apr 25 12:18:44
License: License never expires.

Ty    eq    status  use state ord  capacity    free    mcntg  part   high   low
Ms    10    m-------49% on         1.980G      1.001G      8  16     80%    70%
Md    11    ----------49% on   0   1.980G      1.001G
```

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 “Status Codes” in this chapter.
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.
mcntg	Maxcontig (maximum number of contiguous blocks) parameter.

Field	Description
high	High disk usage threshold percentage.
low	Low disk usage threshold percentage.

(e) - Staging Display

The staging status display shows the status for removable media I/O activity. To view the staging status display, press the n key. To view the I/O status for all removable media devices, enter :n. To view the status for a specific device, enter :n *media*, where *media* is the media type. Press CTRL-k to list the file path name on the second line of each entry.

Sample Display

The following figure shows a sample display of the staging status for two types of media, xt and at:

```
Removable media I/O activity: all      samu   3.5.0 Tue Apr 25 12:21:38
License: License never expires.        count: 3

Ty    eq    pid    file_size    staged_in    fseq   ino   position    offset
Xt    100   0              0            0       0   0     0           0
At    201   0              0            0       0   0     0           0
At    202   0              0            0       0   0     0           0
```

Field Descriptions

Field	Description
ty	Device type.
eq	Equipment ordinal of the drive.
file_size	Size of the file in bytes.
staged_in	Number of bytes of the archive file that have already been staged to disk.
ino	Inode number
position	The position (in decimal) of the archive file on the specific medium.

Field	Description
offset	Offset of the archive file on the specific medium.

(f) - 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

```
Optical drive status:            samu   3.5.0   Fri Feb  9 11:20:33
Ty      eq      status           act    use     state   vsn
Mo      35      --l---wo-r       1      29%     ready   oper2
```

Field Descriptions

Field	Description
ty	Device type.
eq	Equipment ordinal of the optical disk.
status	Device status. For a description of status codes, see Status Codes in this chapter.
act	Activity count.
use	Percentage of cartridge space used.
state	Current operating state of the optical disk. Valid device states are as follows: ready - the device is on and the disk loaded in the transport is 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 nolabel if the volume is not labeled.

(g) - 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.

To view load requests for all removable media, enter :p.

To view load requests for a specific type of medium, enter :p *dt*, where *dt* is the type of medium.

To select either the manual/robot display or the priority display, press the CTRL-k key sequence.

Sample Displays

Removable media mount requests all both samu 3.5.0 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

Removable media load requests all prioritymu 3.5.0 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

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.

Field	Description
priority	Priority of the request.
rb	Equipment ordinal of the robot in which the requested VSN resides.
flags	Flags for the device. See the Flags table that follows this table.
wait	The elapsed time since the mount request was received.
count	The number of requests for this VSN if a stage.
vsn	Volume serial name of the volume.

The flags from the preceding table's flags field are as follows:

Flag	Description
W------	Write access requested
-b-----	Entry is busy
--C----	Clear VSN requested
---f---	File system requested
-----S-	Flipside already mounted
------s	Stage request flag

(h) - Removable Media Status Display

The removable media status display allows 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

```
Removable media status: all          samu   3.5.0 Tue Apr 25 12:43:31
License: License never expires.
```

```
Ty      eq      status          act     use     state   vsn
Xt      100     --l------r0     0%      ready   000000
                Labeled Fri Mar  3 09:07:08 2000, blocksize 16384
At      201     ---------p      0       0%      notrdy  empty
at      202     ---------p      0       0%      notrdy  empty
```

Field Descriptions

Field	Description
ty	Device type.
eq	Equipment ordinal of the drive.
status	Device status. For a description of status codes, see “Status Codes” in this chapter.
act	Activity count.
use	Percentage of cartridge space used (optical disk only).
state	Current operating state of the removable media. Valid device states are: 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 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. nolabel if volume is not labeled. Blank if no volume present in the transport, or the device is off.

(i) - Device Status Summary Display

The device status summary 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

```
Device status          samu   3.5.0 Tue Apr 25 12:46:43
License: License never expires.

Ty     eq state        device_name               fs status   pos
Ms     10 on           samfs1                    10 m---------

Md     11 on           /dev/dsk/c0t0d0s0         10 ----------

Xt     100 on          /dev/rmt/0cbn             --l------r
    Labeled Fri Mar  3 09:07:08 2000, blocksize 16384

Sl     200 on          /dev/samst/c1t3u0         200 m--------r
    move complete

at     201 on          /dev/rmt/1cbn             200 ---------p
    empty

at     202 on          /dev/rmt/2cbn             200 ---------p
    empty

ss     300 off         /etc/fs/samfs/rmt300       300 ---------r

hy     301 on          historian                  301 ----------
```

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. In the preceding sample output, it is samfs1.
fs	Equipment ordinal of the family set to which the device belongs.
status	Device status. For a description of status codes, see “Status

Field	Description
	Codes" in this chapter.
pos	Device position.

(j) - Tape Status Display

The tape 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

```
Tape drive status                    samu  3.5.0  Tue Apr 25 14:41:03
License: License never expires.

Ty     eq     status        act     use     state    vsn
Xt     100    --l------r0   0%      ready   000000
       Labeled Fri Mar  3 09:07:08 2000, blocksize 16384

At     201    ---------p    0       0%      otrdy
       Empty

At     02     --------p0    0%      notrdy
       empty
```

Field Descriptions

Field	Description
ty	Device type.
eq	Equipment ordinal of the tape device.
status	Device status. For a description of status codes, see "Status Codes" in this chapter.
act	Activity count.
use	Percentage of the volume that has been used.
state	Current operating state of the tape device. Valid device states are as follows: ready - the device is on, and the tape loaded in the transport is

Field	Description
	available for access.
	notrdy - the device is on, but no 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 tape. nolabel if the tape is not labeled.

(k) – Display All Files in Staging Queue

The samu 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.

The following subsection contains a screen snap of the samu(1M) utility's u display.

Sample Display

```
Stage queue for media type: all        samu  3.5.0     Fri Dec 10 16:46:58
 License never expires                        files queued: 7
 slot ty   eq    pid  file_size  fseq       ino   position     offset
    1 lt   23  26754   287.241k     4       240 0x0000358d 0x00005cef
      /sam4/testdir0/filenl
    2 lt   23  27333   617.513k     4       241 0x0000358d 0x00005f31
      /sam4/testdir0/filenk
    3 lt   23  27333   137.458k     4       242 0x0000358d 0x00006407
      /sam4/testdir0/filenj
    4 lt   23  27333     1.405M     4       243 0x0000358d 0x0000651c
      /sam4/testdir0/fileni
    5 lt   23  27333     1.613M     4       244 0x0000358d 0x0000705b
      /sam4/testdir0/filenh
    6 lt   23  27333     1.558M     4       245 0x0000358d 0x00007d45
      /sam4/testdir0/fileng
    7 lt   23  27333   231.942k     4       246 0x0000358d 0x000089bf
     /sam4/testdir0/filenf
```

Field Descriptions

Field	Description

Field	Description
ty	Device type.
eq	Equipment ordinal of the tape device.
pid	UNIX process identifier. A process identifier of 1 indicates NFS access.
file_size	Size of file in bytes.
fseq	File system equipment number.
ino	inode number
position	The position (in decimal) of the archive file on the specific medium.
offset	Offset of the archive file on the specific medium.

(l) - Robot VSN Catalog Display

The robot VSN 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. When 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 2-line display that shows the times and barcodes.

Sample Display

```
Robot VSN catalog by slot      : eq 200  samu  3.5.0 Tue Apr 25 14:47:55
License: License never expires.                         count 12
slot         access time count use flags         ty vsn
   0    2000/04/25 13:44    0   0% -il-o-b-----  at 002939
   1
   2    2000/04/25 13:44    0 100% -il-o-b-----  at 002936
   3    2000/04/25 13:45    0  32% -il-o-b-----  at 000143
   4    2000/04/25 13:45    0  62% -il-o-b-----  at 000148
   5    2000/04/25 13:47    0  32% -il-o-b-----  at 002938
   6    2000/04/25 13:47    0   0% -il-o-b-----  at 000159
   7    2000/04/25 13:48    0   0% -il-o-b-----  at 000014
   8    2000/04/25 13:48    0  36% -il-o-b-----  at 002937
```

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.
use	Percentage of space used for the volume.
flags	Flags for the device. See the Flags table that follows this table.
ty	Device type.
vsn	Volume serial name of the volume.

In some cases, more than one flag can occur in a field, and one flag overrides the other. The flags for this table are as follows:

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--------	Bad media. Set when ASM or ASM-QFS detects a write error on a cartridge.
----o-------	Slot occupied.

-----C------	Volume is a cleaning tape. Overrides p.
-----p------	Priority VSN.
------b-----	Barcode detected.
-------W----	Write protect. Set when the physical write protection mechanism is enabled on a cartridge.
--------R---	Read only.
---------c--	Recycle.
----------d-	Duplicate VSN. Overrides U.
----------U-	Volume unavailable.
-----------X	Export slot.

Chapter 10 - How to Upgrade Your Environment

This chapter describes how to upgrade the hardware within your existing ASM or ASM-QFS environment. The following topics are presented:

- How to add slots in an automated library
- How to upgrade or replace an automated library
- How to upgrade DLT tape drives

Certain other types of operations and upgrades also need to be performed within an ASM or ASM-QFS environment. The following ASM publications describe these other types of upgrades:

- The ASM File System Administrator's Guide describes the following types of operations and upgrades in chapter 4, "File System Operations":
  - How to make a file system
  - 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 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 server
  - How to upgrade your Solaris operating system in a QFS, ASM, or ASM-QFS environment
- The ASM Installation and Configuration Guide describes upgrading QFS, ASM, and ASM-QFS software.

How To Add Slots in an Automated Library

The number of cartridge slots managed by the ASM or ASM-QFS system is controlled by the ASM license key. To increase the number of slots, follow these steps:

1) 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.

2) Replace the existing license keys with the new license keys. License keys start in column one /etc/opt/LSCsamfs/LICENSE.3.5. No other keywords, host ids, or other information can appear. The license becomes effective the next time sam-initd is started.

3) Unload the library catalog using 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. Catalog information can include the number of times a cartridge has been mounted and available space.

   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.

4) Bring down the ASM or ASM-QFS system. For information on how to perform this step, see chapter 2, "Basic Operations". Power down the server and library according to the manufacturer's suggested procedure.

5) Have the library hardware engineer add slots to the automated library. Power on the system using your normal startup procedure.

6) Start the ASM or ASM-QFS system. For information on how to perform this step, see chapter 2, "Basic Operations". The new license information appears in the samu(1M) utility's l display.

How To Upgrade or Replace an Automated Library

Prior to disconnecting and installing a different automated library, prepare for the upgrade as described in the “Preparing for Hardware Upgrades” subsection.

To replace or upgrade an automated library, follow these steps:

1) Unload the library catalog using 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. Catalog information can include the number of times a cartridge has been mounted and available space.

   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.

2) Update the /etc/opt/LSCsamfs/inquiry.conf file, if needed. The new library should be identified in this file by the vendor, the automated library model, and by an ASM or ASM-QFS internal name.

   For example, the released inquiry.conf file has 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 sam-init is re-started.

3) Save the current /etc/vfstab file as /etc/vfstab.cur. Edit the /etc/vfstab file changing any ASM or ASM-QFS mounts from yes to no.

4) Save the /etc/opt/LSCsamfs/archiver.cmd file as archiver.cmd.cur. Edit the /etc/opt/LSCsamfs/archiver.cmd file adding a wait directive as the first line.

5) Power down the server and peripherals using the manufacturer's suggested procedure. Disconnect the automated library.

6) Attach the connecting cable to the new automated library and power on the peripherals and server using the suggested power-on sequence.

7) Ensure that the server 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.

   Following the probe-scsi-all, enter the following command to boot with the new configuration:

   > boot -rv

8) If the target numbers of the drives or automated library change, or if the ordering or number of the drives in the automated library changes, make sure you modify the /etc/opt/LSCsamfs/mcf file to reflect the new configuration. This is similar to an initial installation as described in the *ASM Installation and Configuration Guide*.

   If adding new equipment, you may need to create new /dev/samst entries. Enter the following:

   server# **samdev**

9) After the appropriate mcf changes and /dev/samst entries are complete, initialize the ASM or ASM-QFS system by either mounting a file system or by entering the following command:

   server# 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 the audit completes without problems, you can replace the /etc/vfstab and /etc/opt/LSCsamfs/archiver.cmd files with the pre-upgrade versions. Use the saved /etc/vfstab.cur and /etc/opt/LSCsamfs/archiver.cur files, respectively. Then reboot the system to ensure that no errors exist in the configuration.

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/LSCsamfs/mcf file. Remember, 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.

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. Perhaps the best 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.

How 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 an 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 for an upgraded license.

The following restrictions and general information should be noted prior to upgrading drives:

- ASM and ASM-QFS environments do not support mixed DLT tape drives within the same SCSI-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 of the DLT drives with the new drives at the same time.

- The lower density tapes can co-exist 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 wish to recycle existing files and migrate them to a higher density tape. You can accomplish this by marking all of the lower density tapes as read only, then marking these tapes to be recycled. For information on recycling tape, see chapter 7, “The Recycler”.

- As tapes are labeled, the density of the tape is acknowledged and recorded in the library catalog.

To upgrade tape drives, follow these steps:

1) As a precaution, run samfsdump(1M) on your ASM or ASM-QFS systems.

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 by an 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/LSCsamfs/examples/st.conf_changes. You can read the entire file in to /kernel/drv/st.conf or merge the necessary changes. For more information on updating the st.conf file, see the *ASM Installation and Configuration Guide*.

3) Power down the server and peripherals using the manufacturer’s suggested procedure. Replace the tape drives with the new drives.

4) Power on the peripherals and server using the suggested power-on sequence.

5) Ensure that the server 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.

Following the successful probe-scsi-all command, enter the following command to boot with the new configuration:

> boot -rv

6) If the target numbers of the drives or automated library change, or if the ordering or number of the drives in the automated library changes, modify the /etc/opt/LSCsamfs/mcf file to reflect the new configuration. This is similar to an initial installation as described in the *ASM Installation and Configuration Guide*.

   New /dev/samst entries may need to be created for the new equipment. To create these entries, enter the following command:

   server# **samdev**

7) Start your ASM or ASM-QFS system by using your normal startup procedure and mount the file systems. At this time, you can continue to use the existing ASM or ASM-QFS tapes.

Chapter 11 - Advanced Topics

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

Device Logging

The device logging facility provides device-specific error information that can be used 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/LSCsamfs/devlog. The name of each individual log file is the same name as the equipment ordinal.

Example. Assume an ASM file system with a single HP optical library with two optical drives would look like this.

The mcf file is as follows:

```
/dev/samst/c1t5u0    40    hp    hp40  -  etc/opt/LSCsamfs/hp40_cat
/dev/samst/c1t4u0    41    mo    hp40  -
/dev/samst/c1t6u0    42    mo    hp40  -
```

The /var/opt/LSCsamfs/devlog file is as follows:

4-jafs: **pwd**

/var/opt/LSCsamfs/devlog

5-jafs: **ls**

40 41 42 43

6-jafs:

Note that device 43 is the historian.

Enabling the Device Log

There are two ways to enable the device log.

Method 1 uses the samset(1M) command, as follows:

server# **samset devlog** *eq event*

Method 2 requires a directive to the /etc/opt/LSCsamfs/defaults.conf file. Edit the defaults.conf file and add the following directive:

devlog *eq event*

For both method 1 and method 2:

- eq is the equipment number of the device from the mcf file or the keyword all for all equipment.
- event is one or more event types from the following list:

all	none	default	stage_ck
detail	err	label	
msg	mig	retry	
stage	syserr	time	
module	event	date	

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

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.

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 default:

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 may 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 also automatically collected when info.sh is invoked. This allows the file system service to review any possible device error information as part of problem analysis activity.

Logging Events

The device logging events are listed in the samset(1M) man page. They are also listed in a preceding subsection, “Enabling the Device Log”. Note that the device log messages are only available in English text.

Removable Media Files

The request(1) command is used 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 normal ASM or ASM-QFS files in that they have permissions, user name, 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 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 inodes file. 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” in this chapter). The ASM or ASM-QFS file system automatically mounts the requested volume if the volume resides in an automated library defined in the master configuration file.

The volumes used for the request 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(1M) expects that archived files are the only usage of 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 using 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 chapter 4, “Archiver Operations”.

Volume overflow files can be created directly by using the request(1) command. Note that using the request(1) command bypasses the normal 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.

<u>Example 1.</u> The following is an example request(1) command that creates a removable media file on Ampex D2 tapes using three volumes:

server# **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:

server# **sls -D file50**

```
file50:

        mode: -rw-rw----  links:   1  owner: gmm    group: sam

        length: 1500000172

        archdone;

        copy1: ---- Jan 13 15:55 lt

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're using volume overflow, and a file you are retrieving spans multiple volumes, see the examples in appendix B, "Control Structures, Disaster Preparation, and Recovery", for information on how to retrieve such a file.

Segmented Files

The ASM and ASM-QFS environments support segmented file systems. Segmented file systems improve 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.

The segment(1) command allows you to specify the segment size. Residence depends on the size of the disk cache.

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 can be 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 file systems support all other ASM and ASM-QFS capabilities.

The following subsections describe characteristics of segmented files that differ from those of nonsegmented files. For more information on segmented files, see the segment(1) or the sam_segment(3) man pages. Note that the segmented file features are enabled through the ASM-Segment license key.

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.

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

–minsize 20M directive, this file is never archived to that archive set because arfind(1M) selects based on segment size, not file size.

Only segments that have been modified are archived – not the entire file. Up to four archive copies can be made for each segment.

File system data includes directories, symbolic links, the indexes of segmented files, and removable media information. The index of a segmented file contains no user data, so it is assigned to the file system archive set.

Disaster Recovery

In the event of a disaster, successful recovery of a segmented file requires the following:

- Knowledge regarding whether or not the file in question is a self-contained file or a file segment. For information on recovery of self-contained files, see appendix B, "Control Structures, Disaster Recovery, and Preparation". The information in appendix B is useful in recovering individual file segments, but the information in this subsection describes disaster recovery that is specific to file segments.

- The segment size. This is the size of the first file segment.

The remainder of this subsection is an example showing how to recover a segmented file named aaa.

The following is recorded in the archiver.log file for segmented file aaa:

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

In the preceding archiver.log file, the S in the second-to-last field indicates that the entry is a segment of segmented file aaa.

The first segment of file aaa is ./seg/aaa/1. As a part of the recovery process, you need to specify the segment(1) command. This command sets the segment attribute for an existing file. The segment size must be specified on the segment(1) command line. If a file is segmented, it is archived and staged in segment-sized chunks.

The following commands are the ones used to reassemble segmented file aaa:

```
server# Create a removable media file that points to the segments:
server#
server# request -p 0x1276a -m ib -v E00000 /sam3/rmfile
server#
server# Read the segments from tape into disk cache:
server#
server# star xvbf 512 /sam3/rmfile
seg/aaa/1
seg/aaa/2
seg/aaa/3
server# cd seg
/sam3/seg
server# ls –l
total 8
drwxrwx---   2 root     other       4096 Jun 15 17:10 aaa/
server# ls -l aaa
total 40968
-rw-rw----   1 root     other    10485760 Jun 15 17:06 1
-rw-rw----   1 root     other    10485760 Jun 15 17:06 2
-rw-rw----   1 root     other         184 Jun 15 17:07 3
server# pwd
/sam3/seg
server# cd aaa
/sam3/seg/aaa
server#
server# Sort the numbered files in numerical order and cat(1) them together
to create
server# file bbb, which is not segmented:
server#
server# ls | sort -n | xargs cat > ../bbb
     # server# cd ..
/sam3/seg
server# ls –l
total 41000
drwxrwx---   2 root     other       4096 Jun 15 17:10 aaa/
-rw-rw----   1 root     other    20971704 Jun 15 17:11 bbb
server# ls -l aaa
total 40968
-rw-rw----   1 root     other    10485760 Jun 15 17:06 1
-rw-rw----   1 root     other    10485760 Jun 15 17:06 2
-rw-rw----   1 root     other         184 Jun 15 17:07 3
server# rm -rf aaa
server#
server# Use the touch(1) and segment(1) commands to create
server# a segmented 10-megabyte file:
server#
```

```
server# touch aaa
server# segment -l 10m aaa
server# cp bbb aaa
server# rm bbb
server# sls -2K aaa
-rw-rw----   1 root     other     20971704 Jun 15 17:12 aaa
---------- ----- sI {3,0,0,0}
-rw-rw----   1 root     other     10485760 Jun 15 17:12 aaa/1
---------- ----- sS
-rw-rw----   1 root     other     10485760 Jun 15 17:12 aaa/2
---------- ----- sS
-rw-rw----   1 root     other          184 Jun 15 17:12 aaa/3
---------- ----- sS
```

Appendix A - Requesting Help from Software Support

NOTE

Information for this Appendix was taken from *StorageTek's Software Support Manual*, Part Number 112124005.

In this Appendix, "software problem" refers to problems with and/or questions concerning both the software and the supporting documentation.

About This Appendix

This Appendix outlines the customer Software Support services provided by StorageTek's technical support organizations, and details the process and guidelines for getting help from StorageTek for problems with StorageTek software.

Contacting Software Support and Other Services

Around the world, StorageTek Software Support experts are committed to helping you use your software as effectively as possible. StorageTek can be contacted for software support problems via the following methods:

U.S. and Canada

Telephone	800-678-4430 or 303-678-4430
Fax	303–673–8015 or 303–661–4772
World Wide Web	http://www.storagetek.com or www.support.storagetek.com
E–mail	softsupt@louisville.stortek.com

Learning Products Organization

Telephone	303–673–6262
World Wide Web	http://wfd.stortek.com/

International

To obtain a list of International Subsidiary and Distributor Locations and phone numbers, use the following URL:

World Wide Web	http://www.storagetek.com/global

International customers should contact their local Distributor or Subsidiary office for software support. Additionally, international customers may contact the U.S. Software Support center for assistance in the event that local Distributor or Subsidiary support cannot be reached.

Hours of Operation

Software Support operates from StorageTek headquarters in Louisville, Colorado. StorageTek provides software support seven days a week, 24 hours a day, 365 days a year.

Eligibility for Software Support

You are entitled to receive support for all software products for which there is a current maintenance contract. StorageTek's Software Support provides assistance for software problems or for answering questions based on your contract.

The hours that you can call Software Support are based on this contract. For example, you may choose to contract for Software Support services only between the hours of 8:00 a.m. – 5:00 pm, Monday through Friday, or contract for the full coverage of seven days a week, 24 hours a day, 365 days a year.

Determining Technical Severity for Problem Reports

Severity codes are assigned to each software problem reported to Customer Support. The following guidelines are used by Software Support in assigning a severity level to your problem.

Severity Level 1–The software product is non-operational, resulting in a critical system condition requiring immediate resolution. Support personnel will require continuous availability of a qualified customer technical contact until a circumvention or resolution is provided.

Severity Level 2–The software product is operational, but with severely restricted functionality or system degradation.

Severity Level 3–The software product is operational, with functional limitations or restrictions that are not critical to the overall system operations.

Severity Level 4–Questions associated with product usage, implementation, performance, or any other inquiries for the Software Solutions Support Organization.

NOTE
Software Support may downgrade or upgrade a severity code after analysis of your software problem.

Customer Resource Center (CRC) Web Page

Through the CRC (http://www.support.storagetek.com), StorageTek has developed the following Web tools to assist you in being more productive, as well as assist you in resolving problems or answering your product and application questions.

- Software support through the Online Request Form
- Online software product information
- Online product user guides
- Frequently asked questions and technical tips
- Online software fixes
- Online ordering of software documentation

Online customer training information and registration

Providing Help and Technical Assistance

This section defines how Software Support operates to achieve customer satisfaction.

Scope of Support

The goal of Software Support is to assist you with the use of StorageTek software by answering questions and resolving problems specifically related to the software. The underlying assumption is that you are generally knowledgeable about the product but have additional questions or problems not answered in the documentation or training.

However, there are several situations which are beyond the scope of Software Support. Some examples are:

- Your need for product training
- Code or scripts that you have written or modified
- Fixes to prior releases of software versions that are no longer supported
- Your computing environments, such as network configuration
- Operations and performance tuning
- Your user code exits

For problems where modified code or user code exits are involved, Software Support will confirm that the linkage between the StorageTek program/subroutine and your code is set up properly. Problems encountered thereafter that are not reproducible in the StorageTek source programs, and are attributed to your application or other software interfaces, are outside the scope of Software Support. You will be referred to our Consulting Services, who can provide assistance on either a fee basis or charge on a time and material basis.

Debugging

In most cases, problems and code defects can be fixed through standard software support. It is possible, however, for Software Support to determine that a product problem is caused by conditions at your site rather than by a defect in the StorageTek product. In this case, continued support can only be provided through Consulting Services on a fee basis or a time and material basis.

Support of Product Releases

StorageTek continues to enhance and modify its products to meet the changing demands and requirements of the markets it serves. StorageTek provides product phone support for all releases of software until we notify you that we are discontinuing support. However, we will only make fixes to reported software problems for the current supported software releases. StorageTek mails notification letters well in advance of the non–support date.

Platform Environments

StorageTek software is warranted to perform as documented on hardware platforms running versions/releases of operating systems that are currently supported by their manufactures.

StorageTek will not assign software support resources to correct technical problems for systems that the manufacturer does not currently support. In these cases, you will be referred to Consulting Services.

Operating System Support Policy

StorageTek supports its software products on operating systems that are:

Currently supported and maintained by hardware vendors, and StorageTek certified products as defined in the StorageTek product documentation.

When an operating system is no longer supported by the hardware vendor, StorageTek will honor current maintenance contracts, and provide ample notice for any operating system for which we will discontinue support

Support of User Modified Code

StorageTek programs that you have modified, such as user exits and script files, will not be supported by Software Support. You will be referred to Consulting Services for assistance with problems related to your modified code.

Product Training and StorageTek Support

StorageTek Software Support has found that it cannot provide product training over the telephone. It is neither effective nor efficient to train this way. We make every effort to devote support time to the exclusive purpose of addressing problem issues and answering specific application questions. Software Support is not a substitute for product training.

StorageTek's Learning Products Organization offers product training for all of its products. If you are unfamiliar with basic product concepts and need product training, you will be referred to the StorageTek Learning Products Organization.

You can contact StorageTek's Learning Products Organization through the following:

Telephone	303–673–6262
World Wide Web	Web http://wfd.stortek.com/

Reporting Your Problem

StorageTek's Software Support provides assistance for software problems or for answering questions that are not covered in the product manuals. This section describes the steps to take in reporting your software problem to StorageTek. You should perform the following steps for each software problem you encounter:

a. Use the Customer Resource Center (CRC).

First, try to resolve the problem by using the CRC located at http://www.support.storagetek.com. The CRC provides you with the following types of resources:

- Product User Guides
- Release Notes
- Frequently Asked Questions
- Maintenance/Fixes

b. Be prepared with information about your problem.

If the online WEB resources do not provide a solution, be prepared to provide Software Support the following information:

Customer Information

Issue number if you are working on an open issue or a recurring problem

- Site location number
- Company name
- Your name and phone number.
- Alternate contact number

System Information

- Software product release version numbers and maintenance level

- Hardware platform and operating system level
- System configuration information, such as peripheral devices and network configuration

Problem Information

- What were you trying to do? For example, what command were you using?
- Were there error messages, application logs and trace files, or system error logs? If yes, what were they?
- Is the problem new or recurring? Can the problem be reproduced? If yes, under what conditions?
- Were there any changes made to the system recently? If yes, did the problem occur prior to the change?

c. Open a software issue with Software Support.

Please have your product manuals available, and be prepared to provide any other information that may help Software Support assist you. Software Support recommends that you designate a single technical contact, by product group, to communicate with StorageTek Software Support.

There are two methods for opening an issue with Software Support:

1) Use the StorageTek Customer Resource Center (CRC) online request form for non–Severity 1 issues. The online request form can be accessed through the web at http://www.support.storagetek.com and clicking Product and CRC Support. Select the Online Request Form.

2) Call Software Support at 1–800–678–4430 or 303–673–4430.

d. Open a separate software issue for each software problem.

Provide Software Support with a detailed description and information about each software problem.

e. Record the assigned issue number for future reference.

All of the problem details reported are entered into Software Support's problem tracking database. A unique issue number is then assigned.

f. Software Support may ask for product specific diagnostic material or additional documentation.

Collect the additional information and use one of the methods described in "Sending Documentation, Error Logs/Trace Files, or Problem Records" on page 15 to send the information to StorageTek's Software Support Center. Always reference the issue number and the support representative's name that requested the information on all supporting items of documentation. For envelopes or packages, the issue number should be noted in the lower–left corner.

g. If, at any time, you are unhappy with the support provided by StorageTek Software Support, please contact Software Support and ask to speak to a Software Support Manager.

Responding to and Resolving Customer Calls

Call response time and problem resolution time are non–related.

- Call response time is the amount of time it takes a StorageTek support representative to return your initial call if your call cannot be taken immediately.
- As a guideline, the support representative will return your call within 15 minutes.
- Problem resolution time refers to the total amount of time between when StorageTek receives your initial call and when StorageTek provides a resolution to your problem.

If we have to call you back, or your problem has been escalated to StorageTek's Engineering Solutions Support Team for advanced diagnosis, the following occurs:

- We will make three attempts to contact you using primary and alternate telephone numbers and contacts.
- Your issue will be closed if after three tries we cannot reach you or your alternate contact and we have not received a callback. Software Support will re–open your issue if you call back after your issue is closed.

Problem Resolution

Some software problems can be resolved immediately when you call Software Support and some problems require more time. While it is our intent to resolve calls as quickly as possible, we cannot project exact resolution times for each call. Resolution times are dependent on the following:

- Complexity of the problem
- Speed with which we receive the requested problem documentation and logs
- Number of calls received by Software Support

Immediate Resolution

A Software Support representative will take your initial software support call. The support representative checks the problem tracking database to determine whether the problem was previously reported by other customers. If the software problem was previously reported and resolved, the support representative may have an immediate answer to your problem. Depending on your product, a work-around or PTF/Patch may be available.

If the problem was previously reported by other customers, but not yet resolved, the support representative links your problem issue to the already existing issue. A separate issue number for your specific software problem is also maintained. When a resolution becomes available, you are then provided the resolution.

When Investigation Is Needed

If your software problem cannot be resolved immediately, the support representative escalates your problem to an Engineering Solutions Support Team (SST). An SST engineer will contact you to further diagnose and resolve your problem. In some cases, problem diagnosis may require that SST have access to your system through a modem or the internet.

The SST engineer takes responsibility for researching the problem and determining whether a solution is available. Depending on your product, a work-around or PTF/Patch may be available. If there is no work–around, PTF/Patch, or resolution available, new code may need to be generated to resolve the issue.

If you are operating on a down level release of software, you may be required to move to the most current release or wait for the next product release to resolve your problem.

Closing Your Issue

Once a proposed issue resolution is provided, StorageTek closes the issue. If you require further explanation of the resolution, contact Software Support.

NOTE

International customers should contact their local software support representative.

Recurring Software Problems

If the issue has been closed with Software Support, yet you experience the same software problem, the issue can be reopened. Give Software Support the issue number of the closed issue and state that the problem has recurred.

If the fix you received creates a new problem, you must open a new issue. Give Software Support the issue number of the closed issue and state that the fix created a new problem. You will be given a new issue number to use for future reference.

Sending Documentation, Error Logs/Trace Files, or Problem Records to Software Support

Use the following methods to send problem information to StorageTek Software Support.

NOTE

International customers should follow instructions given by the local support representative when dealing directly with the local Distributor or Subsidiary.

MVS Software

Use the following procedure for reporting MVS software problems:

1) Place information on an 18–track or 36–track Standard Label tape.

2) Use IEBGENER and/or IEBCOPY to place the datasets on the tape, unless instructed by your support representative to use a different utility.

3) Include the StorageTek problem issue number on any documentation sent.

If you can only send a non–labeled tape, you must also send a copy of the DCB information so that Software Support can offload the tape.

NOTE

Failure to send the DCB information slows down the process of problem determination. **StorageTek no longer returns documentation tapes.**

VM Software

Use the following procedure for reporting VM software problems:

- Place datasets on an 18–track tape using VMFPLC2 format.
- Include the StorageTek problem issue number on any documentation sent.

NOTE

StorageTek no longer returns documentation tapes.

Other Software

Email

Small log/trace files can be emailed to:
ssrhql@blackcat.storagetek.com

NOTE

Be sure to include the StorageTek problem issue number as part of the file name.

FTP

Log/trace files can be FTP'd to:

ftp.storagetek.com
Login: Support
Password: <contact Software Support for password>
Directory = /incoming

Mail or Overnight

To send mail or overnight shipment, such as FED–EX, Airborne, or UPS:

Storage Technology Corp.
Attn: Software Support
issue # ______________
One StorageTek Drive
Louisville, Colorado 80028–4280

Fax

Small logs can be faxed to:

303–673–8015
Storage Technology Corp.
Attn: Software Support
issue # ______________

Appendix B - Control Structures, Disaster Preparation, and Recovery

The ASM and ASM-QFS control structure information consists of directories, symbolic links, removable media files, segmented file indexes, and the inodes. In the event of a disaster, this control structure information is needed in order to restore the file system.

Disaster preparation is essential to your site's operational policies. This appendix section describes steps you can take to prepare for a disaster and the steps you can take to recover from a disaster.

This appendix chapter presents the following topics:

- Control structures
- How to use samfsdump(1M) and samfsrestore(1M)
- How to use the .inodes file
- Disaster preparation guidelines
- Disaster recovery
- Using the archiver log file to recover a file

WARNING

For disaster recovery purposes, StorageTek does not recommend that archive files span multiple volumes. Files using volume overflow may be irrecoverable in a disaster situation if you do not keep the archive log information for your system. Your site should be using volume overflow only after assessing these risks and after completely understanding volume overflow. By default, volume overflow is disabled.

Control Structures

ASM and ASM-QFS file systems consist of files, directories, symbolic links, removable media files, and segmented files. The ASM and ASM-QFS file systems keep track of all files in the .inodes file. Note that the segmented file features are enabled through the ASM-Segment license key.

The archiver(1M) creates copies of the data portion of files on cartridges in your site-specified archive sets. The archiver(1M) archives the directories, removable media file information, segmented file indexes, and symbolic links for a file system in an archive set that matches the file system name.

In a disaster, information that pertains to directories, symbolic links, removable media files, the indexes of segmented files, and the inodes, which make up the control structure, is just as important as the data. Restoring this metadata allows you to bring up the file system quickly, providing access to file data that is now offline.

For example, if you created an ASM file system with a family set name of samfs1, there is also an archive set called samfs1. This archive set encompasses the directory, removable media files, indexes of segmented files, and symbolic link information plus any files that are not included in other archive sets.

Note, however, that the .inodes file is not archived. In the event of a disaster, the control structure information needs to be restored before the data can be accessed. The control structure identifies where the archived data resides on cartridges.

It is important to periodically perform control structure dumps to protect your data from a disaster. StorageTek recommends that you perform full samfsdump(1M) backups once or twice a day with backups of the .inodes file more frequently, perhaps as frequently as once each hour. By dumping control structures (using samfsdump(1M)) on a regular basis, file data can be restored (using samfsrestore(1M)) even if the file has been removed from the file system, as long as the cartridge upon which the archive copy was written has not been erased. Also, by saving the control structure dump and the archive media, files or entire file systems can be moved from one file system to another, or even from one server to another.

The following subsections describe the ASM and ASM-QFS control structure information and introduce two methods used to protect ASM and ASM-QFS file systems. Control structures, file system design, and backup methodologies are presented to help you determine the appropriate disaster recovery practices for your site.

Making Control Structure Back-up Copies for File Systems

You can make back-up copies of file system control structures either by using the samfsdump(1M) and samfsrestore(1M) commands or by using the dd(1M) and sammkfs(1M) commands. Each method has advantages and disadvantages. These are as follows:

- Performing a samfsdump(1M) and samfsrestore(1M). The samfsdump(1M) and samfsrestore(1M) commands create and restore control structure dumps of a current directory. samfsdump(1M) saves the relative path information for each file contained in a complete file system or in a portion of a file system. You can also use these utilities to restore the control structure for an entire file system, a single directory, or a file. The samfsdump(1M) is performed on mounted file systems and processes about 100,000 files per minute, depending on the layout of your file system. This is the preferred method.

- Copying the .inodes file and restoring using sammkfs. You can also make backups for the control structure information by copying the .inodes file using dd(1). The .inodes file keeps track of all file information. Making a copy of this file allows you to restore control structure information directly from the inode information using sammkfs(1M).

Using dd(1) is sometimes faster than issuing a samfsdump(1M) command, but no directory, symbolic link, removable media, or file path information is saved when copying the .inodes file, so directory information (archived in the default file system archive sets) is staged automatically when the file system is mounted. This staging can be time consuming depending on the location of the archive set on the cartridge. Also, you can only restore an entire file system. You do not have the option to restore single directories or files.

NOTE

If you set a lengthy archive age for the file system archive set or if you have no volumes assigned to it, do not back up the control structures by using dd(1) to copy the .inodes file.

Dumping and Restoring File System Control Structures

The following are general guidelines for performing dumps:

- Dumps are performed with the file system mounted. At any given time, certain files are in a condition of needing to be archived (because they are new) or needing to be rearchived (because they have been modified).

An *expired* archive copy exists when a file is modified and a new archive image has not yet been created. This file needs to be rearchived. Prior to the new archive image being created, the old image is marked as expired. The presence of expired files can be seen in the output of the sls -D command.

If a samfsdump(1M) is performed prior to the creation of the new archive copies, the following warning message is displayed for all files that do not have a valid archive copy when samfsdump(1M) is run:

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

Upon restoring the file with samfsrestore(1M), the following message appears indicating that the file is now damaged:

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

In summary, be aware that dumping control structures when there are unarchived files generates a warning message indicating that files will be damaged. This message indicates that the files will not be available in the event that you restore from that dump file. Dumping control structures during a quiet period (a time when files are not being created or modified) is a good idea and can eliminate or minimize the occurrence of these messages.

- Perform dumps on a regular basis. You can run the samfsdump(1M) command as a cron(1) job by creating an entry in the crontab file.
- Dumps are a preventative measure against total hardware failure of the ASM cache disk subsystem. You should perform samfsdump(1M) operations and make copies of the .inodes file in order to guard your system from such a failure.
- Ensure that you dump control structures for all ASM and ASM-QFS file systems. Look in /etc/vfstab for all file systems of type samfs.
- Keep more than one copy of control structure dumps and store them in protected locations.

The following subsections describe how to dump and restore control structures. The first method presented uses samfsdump(1M) and samfsrestore(1M). The second method presented uses the .inodes file.

How to Use samfsdump(1M) and samfsrestore(1M)

The samfsdump(1M) command dumps control structure information. The samfsrestore(1M) command restores control structures generated by samfsdump(1M). For a complete description of the syntax and options available with these commands, see the samfsdump(1M) and samfsrestore(1M) man pages.

The following subsections describe dumping and restoring ASM and ASM-QFS control structures using samfsdump(1M) and samfsrestore(1M).

Dumping Control Structures Using samfsdump(1M)

In addition to dumping control structure information, the samfsdump(1M) command supports dumping unarchived data. The –u option on the samfsdump(1M) command causes unarchived data to be interspersed with the control structure data normally contained in a samfsdump(1M) dump. When using the samfsdump(1M) command, please note the following:

- A samfsdump(1M) dump taken under release 3.5.0 using the -u option cannot be restored at earlier releases of ASM software because of new data structures needed to support inclusion of data that has not been archived.

- A samfsdump(1M) dump taken using the -u option can be very large. The samfsdump(1M) command does not have any tape management or estimations such as those associated with ufsdump(1M). You need to weigh the tradeoffs of space and unarchived data when using the -u option. For more information on these commands, see the samfsdump(1M) and ufsdump(1M) man pages.

The following two subsections describe how to dump control structures using samfsdump(1M). They present both a manual method and an automated method.

Manual Method

To manually create a dump for a file system, or for a directory within a file system, perform the following tasks:

1) Login as root.

2) Change to the mount point for the file system or to the directory that you are dumping.

   server# cd /sam

3) Create a dump file by executing the samfsdump command, as follows:

   server# **samfsdump -f** *dump_file*

   For *dumpfile*, specify a full path name.

Automated Method

To automate your dump procedures, use cron(1M) to create an entry in root's cron(1M) table. The following example entry performs a dump and manages the files within a dump directory. Note that in this example, the xargs argument consists of the letter l and the number 1, and not 11 (one one):

```
10 0 * * * (find /csd.directory -type f -mtime +3 \
-print| xargs -l1 rm -f); cd sam; \
/opt/LSCsamfs/sbin/samfsdump -f \
/csd.directory/sam/‘date +\%y\%m\%d‘)
```

Replace /csd.directory with an existing directory of your choice. This entry causes the commands to be executed each day at 10 minutes past midnight. First, dump files older than three days are removed and a new dump is created in /csd.directory/sam/*yymmdd*.

If you have multiple ASM or ASM-QFS file systems, make similar entries for each. Be sure that you save each dump in a separate file.

Restoring File Systems Using samfsrestore(1M)

This example assumes that you have a dump file (*dump_file*) as illustrated in the previous subsection.

In this example, the entire file system is restored into a new directory called restored_fs. You need to determine the appropriate location to receive the restored file system.

1) Change to the mount point for the file system or to the directory. In the following example, the mount point is /sam:

   server# **cd /sam**

2) Create a new directory and change to that new directory. In the following example, the new directory is restored_fs:

   server# mkdir restored_fs

   server# cd restored_fs

3) Restore the entire file system, relative to the current directory, using an existing dump file.

   server# **samfsrestore -f** *dump_file*

Restoring Single Files and Directories Using samfsrestore(1M)

This example assumes that you have a dump file (*dump_file*) as illustrated in the previous subsection.

You can also use a samfsdump(1M) file to restore the control information for a single file or directory. Perform the following tasks:

1) Make a directory in which to restore the files and change to it, as follows:

   server# mkdir restored_fs

   server# cd restored_fs

2) List the name of the file or directory that you want restored using the samfsrestore(1M) command with the following syntax:

   server# **samfsrestore –t -f** *dump_file file_name*

3) Restore the file, relative to the current directory, using the samfsrestore(1M) command with the following syntax:

   server# **samfsrestore –T -f** *dump_file file_name*

   The *file_name* must exactly match the name of the file or directory as it was listed in the previous step.

How to Use the .inodes File

The following subsections describe how to dump and restore control structures using the .inodes file.

NOTE

Do not use this method if you have defined a lengthy archive age for the file system archive set or if you have no volumes assigned to that set. This method requires that file system data be archived regularly to stage directory information. Also, it is important to put this file system information on media that is separate from the file data to minimize the restore time.

Dumping Control Structures Using the .inodes File

The following two subsections describe how to dump control structures using the .inodes file. They show both a manual method and an automated method.

For both methods, note that you can use a variable block size when using the dd(1M) command. Block sizes must be multiples of 16 kilobytes, for instance bs=16k, bs=128k, and so on. An increased block size increases the performance of the copy.

Manual Method

For backup purposes, you can manually create a copy of the .inodes file by performing the following tasks:

1) Login as root.

2) Copy the file to a backup directory as follows:

   server# **sync**

   server# **dd if=/***sam***/.inodes of=/***home***/***samfs***/***inodes* **bs=512k**

Automated Method

To automate the copy of the .inodes file, use cron(1) to create an entry in the root directory's cron table. The following example entry copies the .inodes file every four hours and manages the files within a directory:

0 0,4,8,12,16,20 * * * (find /csd.directory -type f –mtime;
+3 -print| xargs -l1 rm –f);
/bin/dd if=/sam/.inodes of=/var/sam/samfs1.inodes bs=512k)

Restoring a File System from the .inodes File

The file system is recreated using the copied .inodes file as input to the sammkfs(1M) command. The sammkfs(1M) command constructs the file system and restores the inodes. All the files, directories, symbolic links, and removable media files are offline. To completely restore the file system, you (or the archiver) must traverse the directory structure, which causes directory information to be staged after the file system is mounted.

The steps in the procedure are as follows:

1) Restore the file system using the .inodes file as input to the sammkfs(1M) command. For example:

   server# sammkfs -r /var/sam/samfs1.inodes samfs1

2) Mount the file system using the mount(1M) command. In the following example command, the file system name is samfs1:

   server# mount samfs1

3) Change directories to the mount point, which is /sam in the following example, and traverse the directory structure, either by allowing the archiver to scan the file system or by entering the sfind(1M) command, as follows:

```
server# cd /sam

server# sfind . -type d > /dev/null
```

Disaster Preparation Guidelines

This subsection describes steps you can take prior to a disaster. Following these steps can make disaster recovery easier. For each file system do the following:

- Back up the ASM and ASM-QFS file system server disks according to your site's policy. System back-up policies vary from site to site, so you must define this policy for your specific site.

- ASM and ASM-QFS file systems are not back-up products, but an ASM or ASM-QFS file system can be used to store back-up copies of system disks generated by other back-up products. If system back-up copies are stored in an ASM or ASM-QFS file system, make sure that all files can be restored in the order they are needed. For example, avoid scenarios in which the system must be recovered before the file system in which the system's back-up copies reside, but the back-up copies are required to restore the system.

- Make sure that a copy of the ASM or ASM-QFS software (or a back-up copy of the most recently downloaded and installed distribution) is available for reinstallation. This avoids delays in acquiring the replacement software. The best way to ensure that the same software is running before and after a disaster is to back up the ASM software packages when the system's disks are backed up.

- Make sure that all files and directories to be recovered are being archived. Be familiar with the archiver commands. For more information on archiver commands, see chapter 6, "The Archiver", and the archiver.cmd(4) man page.

- Verify that all files to be recovered are being archived in at least one archive set.

- Archive at least two copies of each file onto two different cartridges. Ensure that each archive copy is being archived to a separate cartridge.

* Physically separate second copies from first copies according to your site policy. Spare copies of cartridges can be removed from an automated library when full or on a periodic basis. Copies should be separated by a site-defined distance, whether they are stored on the other side of the room or at another facility. Remote vaulting can be employed to store data to a remote site so spare copies can remain accessible without reintroduction into the library.

* Metadata for an ASM or ASM-QFS file system consists of directory, symbolic link, removable media, segmented file index, and .inode information. In the event of a disaster, the metadata is just as important as file content. The metadata information should be archived to a separate set of cartridges. Make sure that at least two copies of the metadata are being made. Note that the segmented file features are enabled through the ASM-Segment license key.

  You can dump metadata information by using either the samfsdump(1M) command (preferred) or by using the dd(1M) command on the .inodes file. Regardless of the method you choose, you can use the cron(1M) command to automate this procedure.

  Make sure that the samfsdump(1M) or dd(1M) output is written to a fixed disk that resides outside of the ASM or ASM-QFS environment. For more information on dumping control structures, see chapter 5, "ASM File System Overview", and the man pages for the dd(1M), samfsdump(1M), and cron(1M) commands.

* Quiesce user application writes to the ASM or ASM-QFS file system before dumping metadata. This step is optional, but files written to the file system after metadata dumps begin might not be reflected in the metadata dump. These files might not be archived, and archive copies on cartridges might not be reflected in the metadata dump. Consequently, the files might not be known to the system if the dump is used to restore the file system.

  Files written to the file system or archived behind the metadata dump are picked up during the next metadata dump.

* Quiesce archiving activities before dumping metadata or cycling log files. This step is optional, but it ensures that the file system, the archive log files, the archives, and the other log files are all consistent when metadata dumps are created. This simplifies recovery.

* Make sure that the library catalog files and reserved VSNs files are copied to fixed disks that reside outside of the ASM and ASM-QFS environment.

Library catalogs for each automated library and for the historian (for cartridges that reside outside of the automated libraries) are defined in the master configuration file. The reserved VSNs file is located in /var/opt/LSCsamfs/archiver/ReservedVSNs.

- Enable archive logging in the archiver.cmd file. The archive log file lists all files that have been archived and their locations on cartridges. The archive log files can be used to recover files that were archived since the last set of metadata dumps and back-up copies were created.

  Ensure that archive log files are being written to separate, fixed disks that are outside of the ASM or ASM-QFS environment. Archive log files are typically written to /var/adm/*logfilename* on a system disk.

  Assure that archive log files are cycled regularly by using the cron(1M) utility. You can use the mv(1) command to move the current log file to another file. Processes writing to the log file continue to do so until they complete. The ASM and ASM-QFS systems create a new log the next time a process initiates a new write to the log.

  The archive log files grow over time. Old log files can be deleted after files have been archived, metadata dumps have been taken, and back-up files have been created. You might want to retain old log files indefinitely. You can use the old log files to locate and retrieve mistakenly or intentionally deleted or changed file system files from stale archive copies on cartridges.

- Use a cron(1M) script to run the info.sh(1M) command regularly. The info.sh(1M) command writes a SAM report to /tmp/SAMreport. This diagnostic file contains all the configuration information needed to reconstruct an ASM or ASM-QFS installation from scratch if a complete system rebuild is necessary. Make sure that the SAMreport is written to a fixed disk that is separate from the configuration files and resides outside of the ASM and ASM-QFS environment.

- Create removable media files containing tar(1) images of the archiver log files, the .inodes dump output, the samfsdump(1M) output, the library catalogs, reserved VSNs files, the SAMreport, and any back-up and restore cron(1M) or shell scripts. For information on removable media files, see chapter 14, “Advanced Topics”, or the request(1M) man page.

StorageTek recommends that you create two copies of these files and rotate cartridges according to your site’s policy. ASM provides a companion script as a starting point for accomplishing this task. It is provided for reference only, and it reflects only one of many possible system administration strategies. You are responsible for creating scripts to accomplish these tasks.

- Put a copy of the archiver log files, the .inodes dump files, and the samfsdump(1M) files in the ASM file system so they can be archived. This step is optional. The removable media files contain copies of the data. Additional redundancy for old back-up copies can be achieved through typical archiving.

- Make sure that archive copies of the metadata are written to a separate set of cartridges. Keep a written (non-electronic) listing of the cartridges upon which the removable media tar(1M) files, archived metadata tar(1M) files, and directory structures exist. In addition, keep a written record of your hardware configuration. These lists can facilitate a speedy recovery. You can obtain these lists by using the sls(1) command. It produces detailed listings of all directories containing removable media files, and these listings can be emailed to a particular user. For more information on obtaining file information, see the sls(1) man page.

Disaster Recovery

The following subsections describe the recovery procedure and considerations for unrecoverable files.

For disaster recovery information that is specific to segmented files, see chapter 14, "Advanced Features". Segmented file features are enabled through the ASM Segment license key. ASM-Segment is licensed separately from the ASM or ASM-QFS system. For information on obtaining a ASM Segment license, contact your sales representative, your Authorized Service Provider (ASP), or StorageTek.

The Recovery Procedure

The following steps should be performed after a disaster has occurred.

- Determine the system component that failed. Any system component, software element, ASM file system, or ASM-QFS file system that has not failed should not be recovered. However, you might need to reconfigure the ASM or ASM-QFS file system on a restored system in order to regain access to file systems or to determine whether any file systems have failed.

- Recover any available ASM or ASM-QFS configuration files or archive log files from the system's disks before rebuilding the system.

- Obtain the most recent copy of the samfsdump(1M) output file. This might reside on a removable media file, the ASM server disk, the ASM-QFS server disk, or in the most recent file system archive if not available from any other source. Retrieving data from removable media files requires that you create removable media files in an existing, perhaps new or temporary, ASM or ASM-QFS file system consistent with those created during the last metadata dump. Store the samfsdump output file on a disk that resides outside of the ASM or ASM-QFS environment.

- Compare the age of the most recently available samfsdump(1M) output with the most recently available .inodes dump to determine which restoration method will result in a more current file system.

- Restore all the hardware to proper operation.

- Restore the Solaris operating system and ASM software from back-up copies, if possible. Otherwise, install a new system, the latest patch cluster, and the ASM package from the distribution file or from a back-up copy of the most recent download.

- Compare the restored versions of all configuration files represented in the SAMreport with those restored from the system backups. If inconsistencies exist, determine the effect of the inconsistencies and reinstall the ASM or ASM-QFS file system, if necessary, using the configuration information in the SAMreport.

- If you are using reserved VSN files, restore them to their original locations. Replace the most recent library catalog file copies from the removable media file, from the ASM or ASM-QFS server disk, or from the most recent file system archive copies (which are likely to be slightly out of date).

  If the library catalogs are unavailable, build new catalogs using the provided utility and using the application the library catalog section of the most recent SAMreport as input. Use the newest library catalog copy available for each automated library.

  Note that the ASM and ASM-QFS systems automatically rebuild library catalogs for SCSI-attached automated libraries. This does not occur for ACSLS-attached automated libraries. Tape usage statistics are lost.

- If necessary, prepare the disks (binding RAID, mirroring, and so on) before you create ASM or ASM-QFS file systems.

- If the samfsdump(1M) command's output is the most recent metadata dump file, use the sammkfs(1M) command to make a new file system on the ASM or ASM-QFS disk cache disk. Run the samfsrestore(1M) command and use the location of the dump file as an input argument.

- If the .inodes file copy is the most recent metadata dump file, make a new ASM or ASM-QFS file system. Use the location of the .inodes file as input to the sammkfs(1M) command. Note that the ASM or ASM-QFS file system must be operational, and the cartridges containing the directory structures must be available for loading.

- If all file systems have been restored, the system can be made available to users in degraded mode.

- If the most recent .inodes file was used to restore the file system, run the sfind(1) command against the ASM or ASM-QFS file system to find and stage all the known directory structures.

- Run the sfind(1) command against the ASM or ASM-QFS file system to find and stage all files that were known to be staged prior to the outage.

- Run the sfind(1) command against the ASM or ASM-QFS file system to determine which files are labeled as damaged. These files might or might not be restorable from tape, depending on the content of the archive log files. Determine the most recently available archive log files from one of the following sources:

    1) The removable media file.

    2) The ASM or ASM-QFS server disk.

    3) The most recent file system archive if not available from either of the previous two sources. This source is likely to be slightly outdated.

- Run the grep(1) command against the most recent archive log file to search for the damaged files, to determine whether any of the damaged files were archived to tape since the last time the samfsdump(1M) command was run or since the last time the .inodes dump completed.

- Examine the archive log files to identify any archived files that do not exist in the file system.

- Use the star(1) command to restore files from the archive media and to restore files that have been labeled as damaged.

- Re-implement disaster recovery scripts, methods, and cron(1M) jobs using information from the back-up copies.

Unrecoverable Files

Some files might not be recoverable. Files that resided within the ASM or ASM-QFS file system but were not archived cannot be recovered.

If the samfsdump(1M) method was used to dump and back up metadata, the samfsrestore(1M) command flags files without archive copies as damaged. If the dd(1M) method was used to dump and back up metadata, files without archive copies are not marked as damaged.

ASM and ASM-QFS logs cannot help you figure out the files that were not archived and are therefore lost between the last archiver run and the system outage. You can determine the files that might not have been archived by analyzing the archiver.cmd file for archiving directives and intervals. If all files are eligible for archiving, the archiver.cmd file's contents can be used to determine the age of the oldest unarchived (lost) files.

You can analyze the most recent SAMreport to assure that VSNs were available to archive each archive set's data before the outage. Lack of sufficient VSNs can prevent archiving of data in one or more archive sets.

If you are recovering files straight from a back-up tape in tar(1) format, the files are restored to their locations according to the information on the tape. If any files have been moved since the tape was written, they are restored to their original locations, not their new locations.

Using the Archiver Log File to Recover a File

This subsection shows various methods of reading data from magneto optical and tape archive media both with and without using ASM software. Such information is valuable when creating or executing a disaster recovery plan.

A typical archiver(1M) log file entry consists of a single line with fields delimited by spaces. The following are sample lines from an archiver log file with definitions for each field:

A *yyyy/mm/dd* 10:59:19 mo OPT003 samfs1.1 1a5.1 samfs1 27.1 4096 big/HiRes d 0

A *yyyy/mm/dd* 10:59:19 mo OPT003 samfs1.1 1a5.a samfs1 34.1 4096 big/QT/Srs d 0

A *yyyy/mm/dd* 10:59:20 ib E00000 all.1 110a.1 samfs1 20.5 14971 myfile f 0

Reading left to right, the fields in the previous listing have the following content:

Field	Content
1	Archive activity. A for archived. R for rearchived. U for unarchived.
2	Date of archive action in *yyyy*/*mm*/*dd* format.
3	Time of archive action in *hh*:*mm*:*ss* format.
4	Archive media type. For information on media types, see the media(5) man page.
5	VSN.
6	Archive set and copy number.
7	Physical position of start of archive file on media (tar(1) file) and file offset on 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 written on only 1 volume. Length of section (if file is written on multiple volumes).
11	Name of file.
12	Type of file. d for directory. f for regular file. l for symbolic link. R for removable media file. I for segment index. S for data segment. Segmented file features require a ASM-Segment license.
13	Section of an overflowed file or segment. Segmented file features require a ASM-Segment license.

The position on media information in field 7 can be used with the request(1) command with the –p option to position at the beginning of the tar(1) header for the file. Use hexadecimal notation starting with 0x. For example, to position to the previous file, express the position parameter as -p 0xd2e.

The following is the sls -D output for the file. The additional .b in the archive information directs you to the beginning of the tar(1) header for the exact file. The log file simply shows the beginning of the tar(1) file as a whole. .b is the offset * 512 for the logical offset to the beginning of the tar(1) header for the exact file. In this case the file is 5632 bytes from the beginning of the tar(1) file. The output is as follows:

```
server# sls -Di /sam/dir1/dir3/file0
/sam/dir1/dir3/file0:
  mode: -rw-rw----  links:   1  owner: root      group: other
  length:       468  inode:      161
  copy 1:   May  1 15:41       d2e.b    mo v1
  access:   May  1 16:50  modification: May  1 15:41
  changed:  May  1 15:40  attributes:   May  1 15:44
  creation: May  1 15:40  residence:    May  1 16:50
```

With this output as background information, the rest of this appendix contains information on how to recover this file's data.

NOTE

The invocations of the star(1M) command in this appendix use the t option for demonstrative purposes. If you actually wish to extract the data from the tar(1) files, replace the t option with x. For more information on the star(1M) command, enter the following command:

server# star --help

Example 1. This example uses the log file information, the request(1) command, and the star(1M) command to reference a file on a magneto-optical cartridge. The VSN and position is obtained from the log file and used to construct a request(1) command that creates temporary file /sam/xxx, which is passed to star(1M).

The commands and system output are as follows:

```
server# request -p 0xd2e –m mo –l v1 /sam/xxx
server# star tvbf 2 /sam/xxx
tar: blocksize = 16
-rw-rw----  0/1     2673 May  1 15:41 1996 dir3/dir2/file0
-rw-rw----  0/1      946 May  1 15:41 1996 dir3/dir1/file1
-rw-rw----  0/1      468 May  1 15:41 1996 dir1/dir3/file0
```

The following example uses the log file, the dd(1) command, and the star(1M) command to reference the file:

```
server# dd if=/dev/samst/c0t1u0 bs=1k iseek=3374 of=/tmp/junk
count=10 0 0
dd: read error: I/O error    <---- This is OK!
8+0 records in
8+0 records out
server# star tvf /tmp/junk
tar: blocksize = 1
```

```
-rw-rw----  0/1     2673 May  1 15:41 1996 dir3/dir2/file0
-rw-rw----  0/1      946 May  1 15:41 1996 dir3/dir1/file1
-rw-rw----  0/1      468 May  1 15:41 1996 dir1/dir3/file0
```

Example 2. Assume that the data is resident on tape. Here is an example entry from the archiver's log and the sls -D output:

```
A 96/06/04 10:55:56 lt YYY set_1.1 286.1324f samfs1 770.11 test/file40 0 0
```

The output for DLT is as follows:

```
server# sls -D /sam1/test/file40
/sam1/test/file40:
    mode: -rw-rw----  links:   1  owner: root      group: other
    length:    130543
    offline;
    copy 1:   Jun  4 10:55       286.1324f lt YYY
    access:   May 24 16:55  modification: May 24 16:38
    changed:  May 24 16:38  attributes:   Jun  4 10:55
    creation: May 24 16:38  residence:    Jun  4 10:55
```

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

The following output shows how the log file information, the request(1) command, and the star(1M) command can be used to reference a file on tape:

```
server# request -p 0x286 –m lt –v YYY /sam1/xxx
server# star tvbf 32 /sam1/xxx
-rw-rw----  0/1   132881 May 24 16:38 1996 test/file12
-rw-rw----  0/1   131986 May 24 16:38 1996 test/file13
-rw-rw----  0/1   132045 May 24 16:38 1996 test/file07
-rw-rw----  0/1   130543 May 24 16:38 1996 test/file40
-rw-rw----  0/1   125222 May 24 16:38 1996 test/file95
...
tar: directory checksum error             <--- this is OK
server#
```

You can also recover data from tape without using the request(1) command if the ASM or ASM-QFS file system is running on the system.

Make sure the ASM or ASM-QFS environment does not use the tape drive by setting it to unavail using samu(1M), devicetool(1M), or the unavail(1M) command. Then use the load(1M) command to load the desired volume into the drive. Now rewind the tape as follows. If your tape drive is not /dev/rmt/2, substitute the correct name in the following examples.

NOTE

If the system does not have ASM or ASM-QFS software installed on it, make sure that /kernel/drv/st.conf is correctly configured for the tape drive that you are using. For more information on performing this task, see "Step 4: Add Tape Support to st.conf" in the *ASM Installation and Configuration Guide*.

---

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

The first file is the ANSI label, and it is as follows:

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

Here is yet another way to find the block size of the tape. Examine the line in the previous output that begins with 0000240. Following the characters H D R 2 , are 5 characters (1 6 3 8 4). These 5 characters are the bottom five digits of the block size, in decimal. These characters pertain to the block sizes as follows:

Bottom 5 digits of block size	Block Size	512-Byte blocks for tar(1)
16384	16 kilobytes	32 blocks
32768	32 kilobytes	64 blocks
65536	64 kilobytes	128 blocks

31072	128 kilobytes	256 blocks
62144	256 kilobytes	512 blocks
24288	512 kilobytes	1024 blocks
48576	1024 kilobytes	2048 blocks
97152	2048 kilobytes	4096 blocks

The previous output shows the ANSI label displayed. You can skip over it, however, by using the following command in place of the od(1) command.

server# mt –f /dev/rmt/2cbn fsf 1

The following is the first tar(1) file:

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

The following is the next tar(1) file. Note that all files are archived twice. This is the second copy:

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

The following are two copies of another file:

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

The following is the end of the tape:

server# **dd if=/dev/rmt/2cbn ibs=16k obs=10k conv=sync | star tvf -**
0+0 records in
0+0 records out

```
tar: blocksize = 0
server# mt -f /dev/rmt/2cbn status
Other tape drive:
  sense key(0x13)= EOT   residual= 0   retries= 0
       file no= 5   block no= 0
```

NOTE

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

```
read:  not enough space
```

Correct the block size and try again. The block size for star(1) is specified in units of 512-byte blocks.

Glossary

addressable storage

The storage space encompassing online, nearline, and offline storage that is user referenced through an ASM file system.

archiver

The archive program that automatically controls the copying of files to removable cartridges.

archive storage

Copies of file data that have been created on removable cartridges for long-term offline storage.

audit (full)

The process of reading the VSNs from each cartridge in an automated library. For non-tape 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 robot that moves cartridges to and from the storage slots and the drives.

backup storage

A snapshot of a collection of files for the express purpose of preventing inadvertent loss. A backup includes both the file's attributes and associated data.

block allocation map

A bit map representing each available block of storage on a disk and indicating whether the block is in use or free.

cartridge

The 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.

data device

For a file system, a device or group of devices upon which file data is stored.

data space

The portion of a collection of files that is the actual data information.

DAU (Disk Allocation Unit)

The basic unit of online storage.

The ASM file system uses several sizes. The small DAU is 4 kilobytes (2^{17} 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.

The ASM-QFS file systems support a fully adjustable DAU, sized from 16 kilobytes through 65528 kilobytes. The DAU you specify must be multiple of 8 kilobytes.

device logging

A feature that provides device-specific error information used to analyze device problems.

device scanner

Software within the ASM file system that periodically monitors the presence of all manually mounted removable devices and detects the presence of mounted cartridges that may be requested by a user or other process.

devicetool

An ASM-QFS administrative tool with a graphical user interface for viewing information about and managing individual devices.

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.

disk allocation unit

See DAU.

disk cache family set

The definition for the devices that make up a family set. The name of the disk cache family set is found in the equipment identifier field of the Master Configuration File (mcf file). This is sometimes referred to as a *metadevice* in industry literature. Also see family set.

disk striping

The process of recording a file across several disks, thereby improving access performance and increasing overall storage capacity.

direct access

A file attribute (stage never) designating that a nearline file can be accessed directly from the archive cartridges and need not be staged for online access.

directory

A file data structure that points to other files and directories within the file system.

disk space thresholds

User-defined disk space thresholds that define 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 the pre-defined disk space thresholds.

drive

A mechanism for transferring data to and from a volume.

extent array

The array within a file's inode that defines where each data block assigned to the file is located on the disk.

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 mounted within an automated library.

Also see disk cache family set.

FDDI

Fiber Distributed Data Interface. FDDI is a 100 megabytes-per-second fiber optic LAN.

file system-specific directives

Directives that follow global directives 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.

file system

A hierarchical collection of files and directories.

FTP

File Transfer Protocol. An internet protocol for transferring files between two hosts over a TCP/IP network.

global commands

Commands that apply to all file systems and appear before the first “fs = ” line.

indirect block

A disk block that contains a list of storage blocks. The ASM 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.

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 ASM 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.

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.

LAN

Local Area Network.

library catalog

See catalog.

LUN

Logical Unit Number.

mcf

Master Configuration File. The file that is read at initialization time that defines the device topology within an ASM and ASM-QFS environment.

media

Tape or optical disk cartridges.

media recycling

The process of recycling or reusing archive cartridges with low use (that is, archive cartridges with few active files).

metadata

Data about data. The index information needed to locate the exact data position of a file on a disk. Metadata contains information pertaining to the directory, symbolic link, removable media, segmented file index, and .inodes.

metadata device

A separate device (for example a solid-state disk or mirrored device) upon which 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 disjoint sets of disks to prevent loss from a single disk failure. It is often referred to as shadowing.

mount point

The path to a directory where a file system is mounted.

name space

The portion of a collection of files that identifies the file, its attributes, and its storage locations.

nearline storage

Removable 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 network-attached automated library, such as those from StorageTek, ADIC/Grau, IBM, or Sony, is controlled using a software package supplied by the vendor. The ASM and ASM-QFS file systems interface with the vendor software using an ASM media changer daemon specifically designed for the automated library.

NFS

Network File System. A standard protocol that allows a UNIX file system to be remotely mounted via a network.

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).

partition

A portion of a device.

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 only be performed on zero-sized files. That is, the setfa –l command can only be specified for a file that is size zero. For more information, see the setfa(1) man page.

prioritizing preview requests

A method of assigning priority to archive and stage requests that cannot be immediately satisfied.

RAID

Redundant Array of Inexpensive/Independent Disks. A disk technology that uses several inexpensive disks to reliably store files. It may protect against data loss from a single disk failure, may provide a fault-tolerant disk environment, and may provide higher throughput than individual disks.

recycler

AN ASM and ASM-QFS component that reclaims space on cartridges that is occupied by unused 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

AN 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.

robot

The portion of an automated library that moves cartridges between storage slots and drives.

robottool

AN ASM and ASM-QFS administrative tool with a graphical user interface (GUI) for viewing and managing automated libraries.

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, ASM 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 striping.

RPC

Remote Procedure Calls. The underlying data exchange mechanism used by NFS to implement custom network data servers.

ASM

The ASM 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 ASM (Application Storage 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.

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 copy data.

samfsrestore

A program that restores a control structure dump.

samtool

AN ASM and ASM-QFS administrative tool with a GUI for invoking robottool, devicetool, and previewtool.

SCSI

Small Computer System Interface. An electrical communication specification commonly used for peripheral devices such as disk and tape drives and automated libraries.

SCSI-attached Library

An automated library connected directly to a server using the SCSI interface. These libraries are controlled directly by the ASM or ASM-QFS software by using the SCSI standard for automated libraries.

shared writer/shared reader

The QFS shared reader/shared writer capability allows you to specify a file system that can be shared by multiple servers. Multiple hosts can read the file system while only one host can write to the file system. Shared readers are specified with the –o shared_reader option on the mount(1M) command. The one-writer host is specified with the –o shared_writer option on the mount(1M) command. For more information on the mount(1M) command, see the mount_samfs(1M) man page.

small computer system interface

See SCSI.

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. 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 an ASM-QFS file system and defined in the mcf file as 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.

striping

A data access method in which files are simultaneously written to logical disks in an interlaced fashion.

All ASM file systems allow you to declare either striped or round robin access for each individual file system. The ASM-QFS file systems allow you to declare striped groups within each file system.

Also see the glossary entry for round robin.

super block

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.

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.

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

Allows 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. A logical identifier for magnetic tape and optical disk that is written in the volume label.

WORM

Write Once Read Many. A storage classification for media that can be written only once but read many times.

Index