StorageTek

Application Storage Manager™ (ASM)
System Administrator Guide

Release 3.3.1

About This Guide

This guide describes the Application Storage Manager™ (ASM) release 3.3.1 and is intended for system administrators responsible for setting up and maintaining the software. System administrators are assumed to be knowledgeable about the Solaris operating system, including basic Solaris system administrator tasks such as account creation and performing system backups.

ASM 3.3.1 New Features

ASM System Administrator's Guide, 311248301

This rewrite of the ASM Administrator's Guide, publication 311248302, supports the ASM 3.3.1 release.

This release of ASM includes features that might impact system administration. The documentation has been revised to support the new features as follows:

- This release supports Solaris 2.5, 2.5.1, 2.6, and Solaris 7. Solaris 2.4 is no longer supported by ASM.
- The `mount_samfs` command no longer starts `sam-init` if the system is at run level 0, 1, or 2. It starts `sam-init` if the system is at run level 3 or above.
- Users can now set mount parameters in a new file: `etc/fs/samfs/samfs.cmd`.
- The process of prioritizing preview requests has changed from a simple scheme based on the request time to a flexible scheme that allows users to change request priorities.
- The `directio –D` option is now part of the `setfa` and `sam-setfa` commnds. You can use this option to set the direct I/O attribute for a file or directory. If you set this attribute for a directory, it is inherited.
- There is a new script to add or remove the administrator group after installation: `/opt/LSCsamfs/sbin/set_admin.sh`.
- Two new mount parameters were added: `flush_behind` and `stage_flush_behind`.
- The `dump_cat` command has two new options: "`-o`" and "`-V`". The –`o` option lists media that are no longer present in the catalog. The –`V` option displays flags and times as comments in the same format as the `samu v` display.
- The new `archiver.cmd` command `delay = time` allows you to delay the archive process until robots become ready.

- The ASM group is no longer required. `Sam-init` now runs under root `root` or root `operator_group` if an operator group is installed in `defaults.conf`.

Record of Revision

Version	***Description***
3.0	August 1995 Original printing.
3.0.3	March 1996 Manual updated to reflect release 3.0.3.
3.1	January 1997 Incorporates information for the ASM 3.1 and ASM 3.1.1 releases.
3.2	July 1997 Incorporates information for the ASM 3.2.0 release
3.3	March 1998 Incorporates information for the ASM 3.3.0 release.
3.3.1	June 1999 Incorporated information for the ASM 3.3.1 release.

About This Guide

Introduction

This guide describes the Application Storage Manager™ (ASM) release 3.3.1 Its intended audience is system administrators responsible for setting up and maintaining the ASM software. System administrators are assumed to be knowledgeable about the Solaris operating system, including basic Solaris system administrator tasks such as creating accounts and performing system backups.

Organization

This manual is organized as follows:

Chapter/ Appendix	Description
1	Provides an overview of ASM.
2	Describes the steps necessary to install and configure ASM.
3	Describes the basic operations for direct SCSI-attached removable media libraries and manually mounted devices.
4	Describes how to manage the configuration and operation of removable media libraries.
5	Describes the major features of ASM file systems.
6	Describes the theory and operations of archiver.
7	Describes the functions and operation of the ASM releaser.
8	Describes the functions and operation of the ASM recycler.
9	Describes dumping and restoring ASM control structures.
10	Describes the GUIs available with ASM.
11	Describes the curses-based operator tools available with ASM.
12	Describes the Application Programmer Interface (API) released with ASM.
13	Describes how to upgrade the hardware on your ASM storage server.

14	Describes ASM advanced features and procedures.
A	Describes StorageTek support procedures.
B	Describes disaster recovery techniques for retrieving data.
C	Provides manual pages for the ASM release.

Conventions

The following conventions are used throughout this document:

Typeface	Meaning	Example
command	The fixed-space courier font denotes literal items such as commands, files, routines, path names, and messages.	/etc/fs/samfs/mcf
Boldface Courier	The boldface courier font denotes text you enter at the shell prompt.	`Server#` **`sls -D`**
Italic Courier	Italics indicate variables in a command line. Replace variables with a real name or value.	`#` **`mount`** *`mnt_pt`*

Table of Contents

Chapter 1. Overview

Introduction

The Application Storage Manager™ (ASM) provides cost-effective file management, storage, archive, and retrieval services on the Solaris 2.5, 2.5.1, 2.6, and Solaris 7 (32 bit) platforms. ASM automatically and transparently copies files from on-line disk to less expensive automated storage devices, and restores the files to an on-line disk when they are needed. In addition, ASM automatically maintains on-line disk space at specified percentage usage thresholds.

This chapter provides a technical overview. The following topics are presented.

- ASM capabilities
- The storage devices supported by ASM
- An introduction to the ASM file system
- Removable media features
- Disaster recovery
- The structure of ASM software

ASM Capabilities

ASM provides the following capabilities:

- **Archive**

  Files are automatically copied from magnetic disk cache to archive media.

- **Release**

  Magnetic disk cache is automatically maintained at specified percentage usage thresholds.

- **Stage**

  Files are automatically copied from archive media to magnetic disk cache.

- **Recycle**

  Archive media are cleared of expired archive images and made available for reuse.

Archive

By default, ASM automatically creates one archive copy of all files in an ASM file system. The system administrator can configure the archiver to create up to four archive copies on a variety of media. The archiving process is triggered when disk-based files match a set of criteria configured by the system administrator.

The system administrator can also direct:

- Archive files immediately
- Never archive files (super user only)

Release

Two threshold values are used to manage usage of the on-line magnetic disk cache. When on-line disk-allocated space exceeds the high percentage usage threshold value, ASM automatically begins releasing the magnetic disk space of archived files. On-line disk space occupied by archived files is released until the low percentage usage threshold is reached. The release priority for a file is determined by two site-definable weighting factors for file size and file modification age, as follows:

```
priority = (size * weight_size) + (age * weight_age)
```

The system administrator can also direct to:

- Release files immediately after archiving.
- Never release files (super user only)
- Release files, but retain the first portion of the file on-line. This is a particularly useful feature as some applications, such as filemgr(1), read only the beginning of the file. With the partial release option, a portion of the file remains on disk cache and the remainder of the file is released. Reading the first part of the file still on disk cache will not trigger the staging of the rest of the file back to disk cache from the archival medium.

Stage

When an off-line file is accessed, ASM automatically stages the file back to disk cache. For a sequential read of an off-line file the read operation tracks along directly behind the staging operation, allowing the file to be immediately available to an application without staging the entire file.

The user can also direct to:

- Stage files immediately. There is an option to wait or not wait for the stage to complete.
- Never stage files. Some applications randomly access small records from large files. With the never stage option, the data is accessed directly from the archive media without staging the file on-line.

Error processing of stage requests is handled automatically by ASM. If a stage error is returned, ASM 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 covered by automatic processing include media errors, unavailability of media, unavailability of a robot, unavailability of drives for a specific copy on a master media type, or re-labeling media.

Recycle

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 proportion of expired archive copies and directs moving non-expired copies to different volumes. In those cases where only expired copies exist on a given volume, an administrator-defined action is taken. For example, the volume can be relabeled for immediate reuse or exported to off-site storage, thus keeping an historical record of file changes. Users are unaware of the recycling process as it relates to their data files.

ASM Storage Devices

ASM supports a wide variety of optical and tape storage devices.

The topology of the equipment managed by ASM is defined in a master configuration file, `/etc/fs/samfs/mcf.` This file specifies the devices, robots, and file systems. Each piece of equipment is assigned a unique equipment identifier in the mcf file.

The mcf associates robotic storage devices with robots and disk partitions with file systems. Entries in the mcf differentiate between manually mounted devices and robotic devices.

When possible, ASM uses the standard Solaris disk and tape device drivers. For devices not directly supported under Solaris, such as robot and optical disk devices, StorageTek provides special device drivers in the ASM software package.

ASM Structure

ASM consists of the following components: file system, processes, commands and tools.

File System

The file system module is dynamically loaded on the first mount of an ASM file system. The file system module is eligible for unloading when no ASM file systems are mounted.

ASM Processes

`Sam-init` is the master process. The first mount of an ASM file system automatically starts `sam-init.` However, you can also start `sam-init` from the command line with the samd start command. Only one copy of `sam-init` can be running at one time. The following programs are also used by ASM.

Process	Description
`archiver`	Automatically archives ASM files. This process runs as long as any ASM file system is mounted.
`robots`	Starts and monitors media changer control daemons.
`scanner`	Monitors all manually mounted devices. `Scanner` periodically checks each device for inserted media.
`releaser`	Attempts to release disk space of archived files on ASM file systems until a low water mark is reached. `Releaser` is started automatically when a high water mark is reached on disk cache.
`historian`	Tracks media exported from media changers, and media mounted on manually mounted devices.
`stageall`	Daemon that controls the associative staging of ASM files.
`rpc.sam`	Remote procedure call application programmer interface (API) server process.

The `samd(1M)` command is used to start and stop `sam-init` and associated processes.

User Commands

By default, the archiving, releasing, and staging capabilities are transparent to the user. However, each of these operations can be fine-tuned by user commands as described below. In addition, there are several other commands a user may find useful. All commands are documented using the UNIX `man(1)` command. All ASM manual pages are included in Appendix C.

Command	Description
`archive`	Archives files and sets archive attributes on files.
`release`	Releases disk space and sets release attributes on files.
`stage`	Copies off-line files to disk and sets stage attributes on files.
`sls`	Lists contents of directories. `sls` is based on the GNU version of the `ls` command and contains options for displaying ASM attributes and information.
`sfind`	Searches for files in a directory hierarchy. `sfind` is based on the GNU version of the `find` command and contains additional ASM options such as `–archived`, -offline, -online, and –damaged.
`sequest`	Creates a removable media file.
`setfa`	Sets file attributes.
`ssum`	Sets the checksum attributes on files.

GUI Tools

`libmgr(1M)` is a Java-based graphical user interface for managing robots and media. `Libmgr` will be replacing `samtool(1M)`, the current graphical user interface, in the near future. From `samtool,` you can invoke the following administrator tools:

GUI Tool	Description
`devicetool`	Displays and manages SAM-FS devices.
`previewtool`	Displays and manages pending mount requests.
`robottool`	Displays and manages SAM-FS robots and tracks exported media using the `historian(7).`

For more information on these tools, see Chapter 10, “Window-Based Administration Tools”.

In addition to the GUI tools, there is the curses-based ASM operator utility, `samu(1M).` For information on using `samu`, see Chapter 11, “Operator Information”.

Application Programmer Interface (API)

The API allows ASM requests to be made from within a user application. The requests can be made locally or remotely to the machine on which ASM is running. The API consists of the following:

- `libsamrpc and libsam`--ASM library calls to get file status, to set archive, release, and stage attributes on an ASM file, and to manipulate the catalog for a media library.
- `rpc.sam`--A server process that handles remote requests.

For more information on the API see Chapter 12, "Application Programmer Interface (API)".

System Administrator Commands

ASM provides system management and maintenance commands for the system administrator, all of which are completely documented in the manual pages. The following tables give a brief description of the administrator commands. The commands are grouped by functional usage.

The following commands are used for creating and maintaining ASM file systems:

Command	Description
`notify`	Runs a script that notifies the users when abnormal events occur.
`samd`	ASM utility that starts or stops ASM daemons.
`samfsck`	Checks and repairs an ASM file system.
`sammkfs`	Makes a new ASM file system from disk partitions.
`samfsdump` `samfsrestore`	Creates/restores a dump file of the control structures associated with an ASM file system.
`samgrowfs`	Expands an ASM file system by adding disk partitions.
`samncheck`	Returns the full pathname given the mount point and inode number
`samu`	Curses-based operator interface for ASM. Displays status of devices and allows operator to control removable media devices.

Media Library Commands

The media library commands are used to configure, initialize and maintain robots and devices under the control of ASM:

Command	Description
auditslot	Audits a single slot within a specified robot.
build_cat	Builds a catalog file for a robot.
chmed	Sets or clears library catalog flags and values.
cleandrive	Requests that a tape drive be loaded with a cleaning tape.
dump_cat	Displays the content of the binary catalog file.
import export	Imports or exports media from a robot by placing it in the mailslot.
move	Moves a medium from one slot to another.
odlabel	Labels optical disks for use with ASM.
tplabel	Labels tapes for use with ASM.

Archiver Commands (for archiver daemons and archiver)

The archiver commands control the actions of the archiver daemons and the archiver itself:

Command	Description
archive_audit	Generates an audit of all archived files.
archiver	Evaluates the archiver commands file for completeness and accuracy.
set_admin.sh	Adds or removes permissions for an administrator group.

Specialized Maintenance Commands

The following table lists specialized maintenance commands that are available to the system admininistrator:

Command	Description
`dmpshm`	Dumps the shared memory segments.
`gnutar`	Can be used for disaster recovery purposes when you need to read data from archive tapes.
`info.sh`	Creates a system information file used to troubleshoot ASM.
`itemize`	Catalogs an optical disk robot or jukebox.
`load` `unload`	Load or unload a medium for a specified device.
`research`	Marks archive entries to be rearchived.
`recycler`	Reclaims space used by expired archive copies from ASM removable media.
`samcmd`	Executes ASM `samu (1M)` commands from a script.
`samdev`	Creates symbolic links in the `/dev/samst` directory that point to the actual devices to be used by ASM (similar in function to the UNIX `makedev (1M)` command.
`samset`	Changes or displays variables used in the ASM library.
`setsyscall`	Changes the system call in the ASM library.
`set_state`	Sets the state of an ASM device.
`unarchive`	Deletes archive entries for one or more files.
`undamage`	Marks an ASM archive entry for one or more files or directories as undamaged.
`unrearch`	Marks archive entries to prevent them from being rearchived.

Chapter 2. ASM Installation and Configuration

Introduction

This chapter describes the steps necessary to install and configure ASM. The following topics are covered in this chapter:

- The ASM release packages
- Directories and files
- Installation Overview
- ASM System Preparation
- Installation and Configuration Procedure
- Software Upgrade Procedure

Installation Overview

The ASM installation process is covered in two sections:

1. ASM System Preparation - ASM expects that certain hardware and software system requirements be met before installation. This section lists the system requirements to ensure that your server is ready to support ASM.

2. ASM Installation and Configuration - This step-by-step procedure describes copying and installing the software packages to your server, and configuring the software to match the hardware at your site. This section also describes the steps needed to initialize ASM and procedures for checking the status of your system.

Before installing ASM, you must set up and configured the hardware to be used with ASM. The ASM storage server typically consists of: 1) a SPARC-based server running the Solaris operating system and ASM software; 2) one or more removable media libraries with multiple tape or optical drives; and 3) a disk cache consisting of SCSI disks or RAID devices.

ASM Release Packages

The ASM software packages are released via anonymous FTP and CD-ROM in Solaris `pkgadd(1M)` format. These packages must reflect the Solaris version (2.5, 2.5.1, 2.6, or Solaris 7) for the platform on which you run ASM. See the instructions provided with the software for a complete description of the directories and files provided with the release package.

The following packages are distributed on the release CD-ROM:

Package Name	Description
LSCsamfs	SAM-FS software package. This package is a prerequisite to the following SAM-FS packages.
LSCgui	SAM-FS Java™-based graphical user interfaces
LSCjre	SAM-FS Java™ Runtime Environment
LSCibm	IBM 3494 libraries package (optional).
LSCstk	StorageTek (ACSLS) libraries package (optional).
LSCdst	Ampex tape drive package (optional).

ASM releases are identified using alphanumeric characters and ranked *M.F.B-P*, where the following characters represent the release level identification:

M	Major release identification
F	Feature addition identification
B	Bug fix update identification
P	Patch level identification. 1 through 99 indicates a patch release. A through Z indicates a beta release.

The following examples illustrate the release identification convention. Note that the base release of a first feature release of a major release may not contain a patch level.

ASM 3.3.1	Base release of the first feature release of a major release
ASM 3.3.1-10	Patch level of a base release

Directories and Files

The following tables list the directories and files installed with ASM.

Directory	Description
`/dev/samst`	Device driver special files.
`/etc/fs/samfs`	Configuration files and daemon binaries.
`/opt/LSCsamfs/bin`	User command binaries.
`/opt/LSCsamfs/client`	Files for RPC API client.
`/opt/LSCsamfs/examples`	Various example configuration files.
`/opt/LSCsamfs/jre`	Java™ Runtime Environment for SAM-FS GUIs
`/opt/LSCsamfs/include`	API include files.
`/opt/LSCsamfs/lib`	Relocatable libraries.
`/opt/LSCsamfs/man`	Manual pages.
`/opt/LSCsamfs/sbin`	System administrator command binaries.

File	Description
`/etc/fs/samfs/LICENSE.3.3`	License file. For more information, see step 7 in the "Software Installation Procedure" section in this chapter.
`/etc/fs/samfs/archiver.cmd`	Archiver command file. See `archiver.cmd(4).`
`/etc/fs/samfs/defaults.conf`	Sets various default values. See `defaults.conf(4).`
`/etc/fs/samfs/inquiry.conf`	Vendor and product identification strings for recognized SCSI devices.
`/etc/fs/samfs/mcf`	Master Configuration File. See `mcf(4).`
`/kernel/drv/samst`	Driver for SCSI media changers, optical drives and non-motion I/O for tape drives.
`/kernel/drv/samst.conf`	Configuration file for `samst` driver.
`/kernel/fs/samfs`	File system module.
`/kernel/sys/samsys`	System call module.

Additional information about ASM files can be obtained from the on-line manual pages after they have been installed later in this procedure. A printed version of the manual pages can be found in "ASM Manual Pages", Appendix C.

Modified System Files

The following table lists the system file that is modified during the installation of ASM.

File	Description
/etc/name_to_sysnum	System call information file.
/etc/name_to_major	Maps drive to major number.

ASM System Preparation

This sub-section outlines the system requirements that must be met prior to installation of ASM.

Requirement 1: SPARC Server

ASM runs on the following Sun Microsystems, Inc. SPARC-based processor systems:

- SPARCStation 5 and above.
- Sun Ultra 1 and above.
- Sun Enterprise 1000 and above.

Although not officially supported by StorageTek, ASM has been installed on SPARC clones, as well.

These systems are to be up and running prior to the installation of ASM. The server must be capable of reading the ASM release CD-ROM or be network-attached to another system capable of reading CD-ROMs.

Requirement 2: Verify Disk Cache

ASM requires a disk cache for the creation of the ASM file system. This disk cache allows for the creation and management of data files and directories between on-line disk and removable media. The disk does not require any special formatting nor does it need to have a UNIX file system made on it. Make sure that the disk and partitions that you are using do not contain any existing data. This data would be lost when you make the ASM file system.

The disk must be direct attached to the server using a SCSI controller or fibre-channel. Individual disk partitions can be specified from a disk, or the entire disk may be used as a disk cache. Disk arrays, including those under the control of volume management software, are supported.

Use the format command to see the disks attached to your system. The following example shows three disks attached to a server, one internal disk connected via controller 0 on the first target (`c0t1d0`) and two external disks connect via controller 1 on targets 1 and 2 (`c1t1d0` and `c1t2d0`).

```
Server# format
```

```
0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
/iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,8
00000/sd@1,0

1. c1t1d0 <SEAGATE-ST424-0116 cyl 2604 alt 2 hd 19 sec 84>
/iommu@f,e0000000/sbus@f,e0001000/dma@2,81000/esp@2,80000
/sd@1,0

2. c1t2d0 <SEAGATE-ST424-0119 cyl 2604 alt 2 hd 19 sec 84>
/iommu@f,e0000000/sbus@f,e0001000/dma@2,81000/esp@2,80000
/sd@2,0
```

How much disk cache is needed? This ultimately depends upon the size of the files being managed by ASM, the frequency of usage for the files, the applications using the files, etc. A rough estimate can be determined using the following algorithm:

Disk Cache = Largest File (in bytes) +

((Number of Files + Number of Directories) * 512) +

Amount of space needed for “working” files

Requirement 3: Solaris Operating System

ASM relies on a properly configured Solaris 2.5, 2.5.1, 2.6, or Solaris 7 operating system. Check to see that your server is running one of these levels of Solaris by entering the following:

```
Server# uname -sr
SunOS 5.7
```

SunOS levels 5.*x*.*y* levels correspond to Solaris 2.*x*.*y* levels. The above system is running Solaris 2.7.

Requirement 4: Disk Space

The ASM software requires the following amount of disk space in each directory:

`/` (root) directory	4 Mbytes
`/opt` directory	7 Mbytes

Determine the amount of space by issuing the `df` command as follows:

```
Server# df -k /
Filesystem        kbytes  used    avail capacity   Mounted on
/dev/dsk/c0t1dos0 76767   19826   49271    29%     /
```

```
Server# df -k /opt
Filesystem        kbytes  used    avail capacity   Mounted on
/dev/dsk/c0t1dos4 192423  59006  114177    35%     /opt
```

If there is not enough room for the software under each directory, either remove the existing files under each directory or re-partition the disk to make more space available to each file system. To re-partition a disk, see the "Disk Management" chapter in the Solaris System Administration manual.

Requirement 5: Super User Access

You must have super user (root) access to the system to install ASM. It is possible, however, to define an ASM administrator's group so that members of the group have access to most of the ASM administrator commands without having root permission. See Step 1, "Add the Administrator Group" in the ASM Installation & Configuration procedures in this chapter.

Requirement 6: Solaris Patches

The latest SunSoft recommended patches for the Solaris operating system are required. An up-to-date list of Solaris patches required by ASM is included with the ASM software in a file called, "README". This list can also be accessed via the LSC World Wide Web site (URL http://www.lsci.com/) on the "Services" page. The patches needed at the time this manual was written are as follows:

For Solaris 2.5, you need the following patches:

- sd driver patch (102984-01 or later)
- kernel jumbo patch (103093-06 or later)
- libthread.so.1 patch (103164-06 or later, obsoleted by 103093-14)
- isp driver patch (103936-01 or later)
- le driver patch (103244-03 or later)
- st driver patch (103870-03 or later).

For Solaris 2.5.1, you need the following patches:

- isp patch (103934-06 or later)
- libc, libnsl, nis_cachemgr, and rpc.nisd patch (103612-06 or later)
- kernel patch (103640-05 or later)

- st driver patch (103857-13 or later)
- fas patch (104246-03 or later)

  (only if using a fast wide SCSI device).

For Solaris 2.6, you need the following patches:

- libthread.so.1 patch (105568-11 or later)
- 105181-06 kernel update patch

There are currently no recommended patches for Solaris 2.7.

To determine which patches are installed on your system, enter the following:

```
server# showrev -p | more
```

If the patches listed above are not installed, you need to install them before installing ASM. Patches are provided to Sun maintenance contract customers via CD-ROM, anonymous ftp, and the Sun World Wide Web page (URL http://sunsolve.com/). StorageTek is not authorized to redistribute Sun patches.

To install a patch, mount the CD-ROM or transfer the patch software to your system. Follow the instructions outlined in the Patch Installation Instructions and Special Install Instructions in the README file.

Note: Do *not* install the ASM software (using `pkgadd`) before installing the required patches. ASM modifies the system call information file `/etc/name_to_sysnum` causing the install of Solaris patches to fail. If you have already installed ASM, you should remove the package (using `pkgrm`) prior to installing the Solaris patches. See the section “Software Upgrade Procedure” in this chapter for more details.

Requirement 7: Removable Media Devices

The ASM system should have at least one removable media device for archiving files. This device may be a single tape or optical drive, or you have multiple devices such as the drives within a media library.

ASM supports a wide variety of removable media devices. A list of currently supported drives and libraries is available on the LSC, Inc. World Wide Web page, URL http://www.lsci.com on the “Services” page.

The device that you are using must be attached and recognized by the server. The following are general guidelines for attaching removable media hardware to the server. For explicit instructions on connecting these peripherals to the server, refer to the hardware installation guide supplied with the library and drives. If the removable media device is already connected and communicating with the server, skip to the next requirement.

1. Power down the server for connecting devices. Use the `init(1M)` command as follows:

   ```
   server# init 0
   ```

   This brings the system down to the PROM level. At this point it is safe to power off the server and peripherals. See the documentation from the hardware vendor for proper power on/off sequences.

2. Connect the removable media devices and the disk to be used for disk cache, if you have not already done so. Ensure that the SCSI target IDs are unique for each controller. For example, if you are using the internal SCSI host adapter, the internal disk drive ID is usually 3; therefore, any peripheral connected to the internal bus must not have an ID of 3. (Note that typically, the internal disk drive ID is 3 for Sparcs and 0 for Ultras.)

3. Power on the peripherals and server according to the manufacturer's recommended sequence. Before the server boots, press the Stop key and the A key simultaneously to interrupt the boot process. Then enter the following at the PROM "ok" prompt:

   ```
   ok  probe-scsi-all
   ```

   This command returns a series of entries for each device connected to the system. If the server does not "see" all of the devices (disk drives, tape or optical drives, the media library), you should check the cabling. This is often the problem when devices and controllers are not communicating. Ensure again that each device has a unique target ID. Do not proceed until all devices appear when probed.

   In some instances, SCSI devices may use a target number greater than f7 or devices with a LUN greater than 0. This occurs when using DLT 2700 drives, which use a LUN of 1. If this is the case with your system, you must edit the `/kernel/drv/samst.conf` file, when installed.

   See the note on target and LUN numbers in Step 4, "Add Tape Support to `st.conf`" in the ASM Installation and Configuration procedures.

4. Boot the server using the reconfiguration option as follows:

   ```
   ok  boot -r
   ```

Requirement 8: ASM Release Software & License

If you do not have an ASM 3.3 license key, contact your authorized service provider or StorageTek. (see Appendix A, "StorageTek Product Support and Enhancements"). The following information is required:

- Company PO number

- Company name, address, phone, contact
- Host ID on which ASM is to be licensed. To display the Host ID on your system, use the `hostid(1)` command.

For each media library, you need:

- Vendor name/model of the library and the type of media used in the library.
- Number of slots for the media library/media type.
- StorageTek optional products to be used with this license

The license keys allow the system to run indefinitely unless you were issued a temporary license, or until the system exceeds the number of pieces of media for which you are licensed. When a temporary license expires or you exceed the slot limit of the license, the system is no longer able to mount and unload media, and archive, stage, and release files. Access continues unaffected for files already on disk.

ASM Installation and Configuration Procedure

This section describes the step-by-step procedure for installing and configuring ASM for the first time. Upgrading ASM software on an existing server is described in the "Software Upgrade Procedure" section in this chapter. For most of the procedures in this section, you must be logged in as root.

Step 1: Add the Administrator Group (optional)

By default, the ASM administrator commands may be executed by root only. However, during installation you may supply an administrator group name. This allows members of the administrator group to execute all administrator commands except for `gnutar`, `samfsck`, `samgrowfs`, and `sammkfs`. The administrator commands are located in `/opt/LSCsamfs/sbin`.

If you want to enable an administrator group, choose a group name. The `pkgadd` process (Step 3, "Add the Packages") prompts you for this group name.

Add the administrator group name using your site's procedure and the `groupadd(1M)` command, or by editing the `/etc/group` file. Following is an entry from the group file designating a SAM-FS administrator group. In this example, the **`samadm`** group consists of both the `adm` and `operator` users:

samadm::1999:adm,operator

You can also define an operator group that is allowed access only to the graphical user interfaces `libmgr(1M)`, `samtool(1M)`, `robottool(1M)`, `previewtool(1M)`, and `devicetool(1M)`. This group can be defined in the `/etc/fs/samfs/defaults.conf` file as described in Step 10, “Set Up SAM-FS Default Values”, and the `defaults.conf(4)` manual page.

You can add or remove the administrator group after installing the package. This action performs the same function that occurs when you select an administrator group during the package install. You can also undo the effect of this selection and make the programs in `/opt/LSCsamfs/sbin` executable only by “root” or “bin”. The script is `/opt/LSCsamfs/sbin/set_admin.sh`.

Step 2: Mount the Installation CD-ROM

The ASM release files are available on CD-ROM. Insert the CD-ROM and change the directory to the ASM software files.

Note: We reserve the right to update our installation procedures at any time. For this reason, the installation CD-ROM is provided with a printed set of instructions. These instructions provide up-to-date details on installing the ASM release files to the server.

- Running the Solaris Volume Manager, insert the CD-ROM and enter the following:

```
server# cd /cdrom/cdrom0
```

Step 3: Add the Packages

ASM uses the Solaris packaging utilities for adding and deleting software. As such, you must be logged in as superuser (root) to make changes to software packages. `pkgadd(1M)` prompts you to confirm various actions necessary to install the StorageTek packages.

The ASM packages and all optional products reside in the `/cdrom/cdrom0` directory. To satisfy product dependencies, the `sampkg` must be installed first. Run the `pkgadd` command to install all packages answering yes to each of the questions:

```
server# pkgadd -d sampkg (must be first)
```

When you install `sampkg`, you are asked if you want to define an administrator group. Select “y” to accept the default (no administrator group), or select “n” if you want to define an administrator group. You can also define or remove the administrator group later by running the script `/opt/LSCsamfs/sbin/set_admin.sh`.

```
server# pkgadd -d samstk(optional)
server# pkgadd -d samdst(optional)
server# pkgadd -d samibm(optional)
```

An optional set of graphical user interfaces (GUIs) is available with ASM. These GUIs eventually replace our current tools under `samtool(1M)`. The new GUIs require the installation of a Java runtime environment. Add these packages as follows:

```
server# pkgadd -d samgui(optional)
server# pkgadd -d samjre(optional)
```

This manual is available in both pdf and postscript formats. Add this package as follows:

```
server# pkgadd -d samdoc(optional)
```

Step 4: Add Tape Support to st.conf

Some tape devices are not, by default, supported in the Solaris kernel. If your ASM storage configuration uses any of the devices listed in `/opt/LSCsamfs/examples/st.conf_changes`, you must modify the `/kernel/drv/st.conf` file. The st tape driver configuration file for all supported tape drives is `st.conf`. By modifying this file and using ASM, you enable the normally unsupported drives to work with ASM.

The following drives are not officially supported by Solaris:

- DLT 2000, 4000
- StorageTek Redwood SD-3, Timberline 9490, 9840
- IBM 3590 Magstar, IBM 3570
- Sony DTF, Sony AIT
- Fujitsu M8100

Read the `/opt/LSCsamfs/examples/st.conf_changes` file into `/kernel/drv/st.conf`. Make any changes as specified in the file.

Note on Target and LUN Numbers

In some instances, SCSI devices may use a target number greater than 7 or a LUN greater than 0. (This occurs with DLT 2700 drives, which use a LUN 1.) In this case, you must edit both the `/kernel/drv/samst.conf` and the `/kernel/drv/st.conf` files.

For LUNs 1 through 7, make the following changes:

1. Edit `/kernel/drv/st.conf`. Add the following lines for each target/LUN combination, making the appropriate substitutions. For example, the following uses target 4, LUN 1:

   ```
   name=”st” class=”scsi”

   target=4 lun=1
   ```

2. Edit `/kernel/drv/samst.conf.` Un-comment or add the appropriate lines for each device, as follows:

   ```
   name=”samst” class=”scsi”

   target=4 lun=1
   ```

To use targets 8 through 15, make the following changes:

3. Edit `/kernel/drv/st.conf`. Find the following line:

   ```
   # In case there are wide tape drives, one can use
   these targets.
   ```

   Un-comment the pair of lines for each target following this comment.

4. Edit `/kernel/drv/samst.conf`. Add the following lines for each target/LUN combination, substituting appropriately. For example, the following uses target 9, LUN 2:

   ```
   name=”samst” class=”scsi”

   target=9 lun=2
   ```

If you have added new devices since running `pkgadd`:

5. Run the following command to create the device entries in `/dev/samst`.

```
server# /opt/LSCsamfs/sbin/samdev
```

Step 5: Reboot System

Reboot the server with the reconfiguration option (`boot -r`) now. Changes to the `st.conf`, `samst.conf` and the `/etc/name_to_sysnum` files are enabled at this time. During an initial installation of ASM, failure to reboot the system at this time can cause a system panic. This message also appears during Step 3, "Add the Packages".

```
Although not mentioned elsewhere in the documentation, you
*MUST* reboot this system now.  Failure to do so will
result in a panic when you first mount a samfs filesystem.
The panic is the result of the newly-changed
/etc/name_to_sysnum file's not being read except at boot
time.  Thus, the change needed for samfs is not currently
in effect.

---------------------------------------------------

                   You MUST reboot now!

--------------------------------------------------------
```

Step 6: Set Up PATH and MANPATH

Set up `PATH` statements:

- For users running the SAM-FS user commands (for example, `sls (1)`), add `/opt/LSCsamfs/bin` to the user's `PATH` variable.
- For users running the administrator commands, add `/opt/LSCsamfs/sbin` to the PATH variable.
- To use the SAM-FS manual pages, add `/opt/LSCsamfs/man` to the `MANPATH` variable.
- In the Bourne or Korn shells, edit the `.profile` file, change the `PATH` and `MANPATH` variables, and export the variables. For example:
  ```
  PATH=$PATH:/opt/LSCsamfs/bin:/opt/LSCsamfs/sbin
  MANPATH=$MANPATH:/opt/LSCsamfs/man
  export PATH MANPATH
  ```
- In the C shell, change your `.login` and/or `.cshrc` file. For example, the path statement in your `.cshrc` file might look like this:
  ```
  set path = ($path /opt/LSCsaamfs/bin
  /opt/LSCsamfs/sbin)
  ```

The `MANPATH` statement in your `.login` file might look like this:

```
setenv MANPATH
/usr/local/man:opt/SUNWspro:/$OPENWINHOME/\
share/man:/usr/share/man:/opt/LSCsamfs/man
```

Step 7: License ASM

License keys are required to run ASM and associated products from StorageTek. See requirement 8 in the "ASM System Preparation" section for information on license keys.

ASM uses encrypted license keys. The license keys are encoded alphanumeric strings. You receive one or more license keys depending on the system configuration and the products being licensed.

Starting in column one, place the keys on the first line in `/etc/fs/samfs/LICENSE.3.3.`

Each license key must be on a separate line and all keys must start in column one. The last license key cannot end with a newline character. No other keywords, host ids, comments, or other information may appear in the `LICENSE.3.3` file. The license becomes effective the next time `sam-init` is started.

The license keys allow the system to run indefinitely unless you were issued a temporary license. When a temporary license expires, the system is no longer able to mount and unload media, and archive, stage, and release files. If you exceed the number of slots for which the system is licensed, you cannot import or label media.

Once the system is running, you can view the current license settings from the `samu(1M)` "`l`" (the letter "l") display.

Step 8: Configure System Logging

ASM logs errors, cautions, warnings, and other messages using the standard Solaris `syslog` interface. The ASM facility is by default "local7". Add a line similar to the following example to the `/etc/syslog.conf` file.

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

You can read this line from `/opt/LSCsamfs/examples/syslog.conf_changes`.

Note: This entry is all *one* line and has a TAB character (not a space) between the fields.

After adding this line, create an empty logfile, identify the PID for syslogd, and send the syslogd PID an HUP signal. The following example creates a logfile in `/var/adm/sam-log`.

```
server# touch /var/adm/sam-log
server# ps -ef | grep syslogd
server# kill -HUP syslogd-pid
```

See `syslog.conf(4)` and `syslogd(1M)` for more information. A different logging facility can be set in the `/etc/fs/samfs/defaults.conf` file.

Step 9: Configure ASM

Each ASM storage server configuration is unique. The system requirements and hardware used at each site may vary. ASM supports a wide variety of tape and optical devices, robotic media changers, and disk drives. This section of the system administration guide presents general sample configurations. It is up to the system administrator at your site to set up the specific configuration for your ASM storage server.

To configure ASM devices, create the master configuration file `/etc/fs/samfs/mcf`, which contains a line for each device and/or family set in your configuration. The `mcf` contains information to:

- Identify the disk slices to be used and organize them into ASM file systems.
- Identify the media drives to be used and organize them into media libraries.

Each line of the `mcf` file has the following format:

```
Equipment    equipment    equipment    family     device     additional
identifier   ordinal      type         set        state      parameters
```

Delimit the field in each line with space or tabs. Comment lines start with the # symbol. Use a dash (–) to indicate omitted fields. The following table describes the fields. Refer to the online manual page `mcf(4)` for more information.

Field	Description
`equipment identifier`	Required. If the device is a disk slice, this field is the `/dev/dsk` entry.
`equipment ordinal`	Required. Enter a unique integer from 1 to 32757.
`equipment type`	Required. Enter a two-character mnemonic for the device type. Most equipment can use the generic equipment types of **od** (optical disk), **tp** (tape), and **rb** (robot). See the `mcf(4)` manual page for specific equipment types.
`family set`	Optional. If the device is associated with a family set (i.e., a file system or media library), enter the family set name for this device.
`device state`	Optional. Enter a state for the device.
`additional parameters`	Optional. If the device is a disk slice, this field points to the `/dev/rdsk` entry. If the device is a media library, this field is the path name to a catalog file.

Configuration Check Procedure

1. Start ASM by mounting an ASM file system or using the `samd start` command.

2. Verify the order of the drives.

   If your automated robot or jukebox library device contains more than one drive, the drives defined in the `mcf` must be in the same order as the drives viewed by the library controller. The drive order that is recognized by the media changer controller can be different than the order of the SCSI targets or LUNs.

If your media changer has a front panel:

3. Verify the order in which the drives are recognized by the media changer controller. Check the SCSI target IDs displayed by the control panel of the media changer.

If your media changer does not have a front panel:

4. Verify the drive order by using the following method. Make sure you check *each* drive in a library.

   – Make the drive state unavailable to ASM by:

     Entering `unavail` in the device state field of the `mcf` file to configure the drives as unavailable.

     OR

     Using `libmgr` to select the drives and change the drive state to unavailable.

     OR

     Using the `samu :unavail` command.

5. Load a piece of media into the drive using the `load(1M)` command as follows:

   # **load -vsn** *vsn_name device_num*

   OR

   # **load -slot** *n device_num*

6. Determine if the correct drive responds while under ASM control. For tape drives, enter the following information where *X* is the raw tape device entry in the mcf file:

   # **mt -f /dev/rmt/*X* status**

   The following example of a status message indicates a tape is in the drive.

```
# mt -f /dev/rmt/0 status
DLT 7000 tape drive tape drive:
   sense key(0x2)= Not Ready   residual= 0
retries= 0
   file no= 0   block no= 0
```

7. If the tapes did not load or the drives did not return a status, the drives may not be properly configured in the `mcf`. Make sure the order is correct in the `mcf` and repeat this test.

   After modifying the `mcf`, you must reinitialize the `sam-init` daemon. Use the `samd stop` command to stop `sam-init` and the `samd start` command to start `sam-init`.

   Since optical drives are not shared and do not return a status, you should read the SCSI target IDs displayed on the control panel for your library. The order in which the drive targets are reported should be the order they are configured in the `mcf` file. Refer to the hardware maintenance manual for library for instructions on identifying and setting target addresses.

The remainder of this sub-section includes an example of the steps necessary to set up the `mcf` file on a server. There is an example `mcf` file located in `/opt/LSCsamfs/examples/mcf`.

Example Hardware Configuration

The following equipment is available on the example server:

- Two Seagate-ST15230W 4 Gbyte disk drives used as cache
- One DLT-2700 7-slot DLT mini-loader containing one DLT-2000 DLT drive
- One manually-loaded DLT-2000 DLT drive
- One HP Model C1710T Magneto-Optical library containing two HP C1716 magneto-optical drives
- One manually-loaded HP C1716 magneto-optical drive

This equipment is connected to two SCSI buses with the following SCSI targets.

- The server's internal, single-ended, SCSI bus.

Equipment	SCSI target
Manually-loaded MO drive	2
The Solaris internal hard disk drive	3
Manually-loaded DLT drive	4

- A differential SCSI bus connects to the libraries and cache disk.

Equipment	SCSI target
Cache disks	0 and 1
DLT 2700 robot	3, Unit 1
tape drive	3, Unit 0
HP C1710T robot	2
first optical drive	5
second optical drive	6

Example Disk Cache Configuration

The Solaris `format(1M)` command reports that the disks are partitioned as follows:

```
1. c1t0d0 <SEAGATE-ST15230W-0168 cyl 3974 alt 2 hd 19 sec 111>
/iommu@0,10000000/sbus@0,10001000/QLGC,isp@1,10000/sd@0,0
```

```
Current partition table (original):
```

```
Total disk cylinders available: 3974 + 2 (reserved cylinders)
```

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	0-3499	3.52GB	(3500/0/0)
1	root	wm	3500-3972	487.09MB	(473/0/0)
2	backup	wu	0-3973	4.00GB	(3974/0/0)
3	unassigned	wm	0	0	(0/0/0)
4	unassigned	wm	0	0	(0/0/0)
5	unassigned	wm	0	0	(0/0/0)
6	unassigned	wm	0	0	(0/0/0)
7	unassigned	wm	0	0	(0/0/0)

```
2. c1t1d0 <SEAGATE-ST15230W-0168 cyl 3974 alt 2 hd 19 sec 111>
/iommu@0,10000000/sbus@0,10001000/QLGC,isp@1,10000/sd@1,0
```

```
Current partition table (original):

Total disk cylinders available: 3974 + 2 (reserved
cylinders)
```

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	1000-3973	2.99GB	(2974/0/0)
1	unassigned	wu	0	0	(0/0/0)
2	backup	wu	0-3973	4.00GB	(3974/0/0)
3	unassigned	wm	0	0	(0/0/0)
4	unassigned	wm	0	0	(0/0/0)
5	root	wm	0-999	1.01GB	(1000/0/0)
6	unassigned	wm	0	0	(0/0/0)
7	unassigned	wm	0	0	(0/0/0)

One file system (`samfs1`) is placed on slice 0 of disk c1t0d0 and slice 5 of c1t1d0. Another file system (`samfs2`) is created on slice 0 of disk c1t1d0 and slice 1 of disk c1t0d0.

Begin writing the `mcf` file for this example configuration by defining the file system and its disk partitions. Figure 2-1 shows the file system entries in the `mcf` file.

1. Make an `ms` (a mnemonic for mass storage) entry for the first file system. The name of this file system (`samfs1`) is used later when writing the `/etc/vfstab` entry for the file system and making the file system.
2. Make a series of `md` (for magnetic disk) entries listing the partitions that comprise the `samfs1` file system.

3. Make similar entries for the second (`samfs2`) file system.

Note: Be sure that the `/dev/dsk` and `/dev/rdsk` names on each line reference the same c*n*t*n*d*n*s*n* partition.

```
# Disk cache configuration for 2 file systems: samfs1, samfs2
#
# Equipment       Eq    Eq   Fam.  Dev.    Additional
# Identifier      Ord  Type  Set   State   Parameters
#-----------      ---   --  ------ ------  ------------------
samfs1              1   ms  samfs1
/dev/dsk/c1t0d0s0  11   md  samfs1   on    /dev/rdsk/c1t0d0s0
/dev/dsk/c1t1d0s5  12   md  samfs1   on    /dev/rdsk/c1t1d0s5
#
#
samfs2              2   ms  samfs2
/dev/dsk/c1t1d0s0  13   md  samfs2   on    /dev/rdsk/c1t1d0s0
/dev/dsk/c1t0d0s1  14   md  samfs2   on    /dev/rdsk/c1t0d0s1
```

Figure 2-1 Example mcf File: File System Entries

Caution: If you give the wrong partition names, you risk damaging user or system data. (This is true when creating any type of file system.) Make sure you specify disk partitions that are not used on your system. Do not use overlapping partitions.

How to Identify Peripherals Using /var/adm/messages

When your system boots, a series of messages are placed in `/var/adm/messages`. These messages identify the Solaris hardware path to each of the peripherals on your system. To display information from the latest system reboot, search backward from the end of the file. Each peripheral has three lines. Note that the third line wraps to the next line in this example.

```
Aug 23 11:52:54 baggins unix: samst2: Vendor/Product ID = HP
C1716T
Aug 23 11:52:54 baggins unix: samst2 at esp0: target 2 lun 0
Aug 23 11:52:54 baggins unix: samst2 is
/iommu@0,10000000/sbus@0,10001000/espdma@5,8400000/esp@5,880
0000/samst@2,0
```

The first line displays the vendor and product information that the SCSI peripheral reported to the Solaris kernel.

The second line displays the SCSI bus, SCSI target and logical unit number of the peripheral.

The third line displays the peripheral's hardware path. This path is reflected in the `/devices` directory. Symbolic links (symlink) to the `/devices` directory are set up in the `/dev/samst` and `/dev/rmt` directories.

Matching the symbolic link to the peripheral is the key to configuring ASM. Use the `ls`(1) command with the `-l` option in both the `/dev/samst` and `/dev/rmt` directories to point to the pathname of the peripheral.

Configuring a Manually Loaded MO Drive

The HP Model C1716T is target 2 on the internal SCSI bus. The following information is located in the block of lines in `/var/adm/messages` associated with this device:

```
Aug 23 11:52:54 baggins unix: samst2: Vendor/Product ID = HP
C1716T
Aug 23 11:52:54 baggins unix: samst2 at esp0: target 2 lun 0
Aug 23 11:52:54 baggins unix: samst2 is
/iommu@0,10000000/sbus@0,10001000/espdma@5,8400000/esp@5,880
0000/samst@2,0
```

Change directories to `/dev/samst` and use "`ls -l`" to look for a symbolic link that points to this hardware path:

```
lrwxrwxrwx   1 root     other         88 Aug 23 12:27 c0t2u0
->
/devices/iommu@0,10000000/sbus@0,10001000/espdma@5,8400000/e
sp@5,8800000/samst@2,0:a,raw
```

The ASM samst driver uses the name `/dev/samst/c0t2u0` when referencing the device. Make the following entry in `/etc/fs/samfs/mcf`:

```
/dev/samst/c0t2u0  20  od  -  on
```

This entry contains the device name (`/dev/samst/c0t2u0`), a unique ordinal (`20`), the equipment type of the drive (`od`) no family set definition (indicated by the "–"), and the device state (`on`).

Configuring an MO Library

The HP library has three SCSI devices: the robotic mechanism, and the two MO drives that the robot loads and unloads. Look in `/var/adm/messages` to find the messages for these devices.

```
Aug 23 11:52:56 baggins unix: samst16: Vendor/Product ID = HP
C1710T
Aug 23 11:52:56 baggins unix: samst16 at QLGC,isp0: target 2
lun 0
Aug 23 11:52:56 baggins unix: samst16 is
/iommu@0,10000000/sbus@0,10001000/QLGC,isp@1,10000/samst@2,0

Aug 23 11:52:56 baggins unix: samst19: Vendor/Product ID = HP
C1716T
Aug 23 11:52:56 baggins unix: samst19 at QLGC,isp0: target 5
lun 0
Aug 23 11:52:56 baggins unix: samst19 is
/iommu@0,10000000/sbus@0,10001000/QLGC,isp@1,10000/samst@5,0

Aug 23 11:52:56 baggins unix: samst20: Vendor/Product ID = HP
C1716T
Aug 23 11:52:56 baggins unix: samst20 at QLGC,isp0: target 6
lun 0
Aug 23 11:52:56 baggins unix: samst20 is
/iommu@0,10000000/sbus@0,10001000/QLGC,isp@1,10000/samst@6,0
```

Next, look in `/dev/samst` for the three symbolic links which point to the `/devices` files with the same Solaris hardware paths shown in the `/var/adm/messages` file:

```
lrwxrwxrwx   1 root     other 74 Aug 23 12:27 c1t2u0 ->
/devices/iommu@0,10000000/sbus@0,10001000/QLGC,isp@1,10000/s
amst@2,0:a,raw

lrwxrwxrwx   1 root     other 74 Aug 23 12:27 c1t5u0 ->
/devices/iommu@0,10000000/sbus@0,10001000/QLGC,isp@1,10000/s
amst@5,0:a,raw

lrwxrwxrwx   1 root     other 74 Aug 23 12:27 c1t6u0 ->
/devices/iommu@0,10000000/sbus@0,10001000/QLGC,isp@1,10000/s
amst@6,0:a,raw
```

Make the entries in `/etc/fs/samfs/mcf`.

```
/dev/samst/c1t2u0  30  rb  hp30  on  /etc/fs/samfs/hp30_cat
/dev/samst/c1t5u0  31  od  hp30  on
/dev/samst/c1t6u0  32  od  hp30  on
```

The first line defines the robot itself. It contains the `/dev/samst` name for the device (`/dev/samst/c1t2u0`) followed by a unique ordinal (`30`), the equipment identifier (`rb`, for a generic library), the *family set* identifier specified on all devices associated with this library (`hp30`), and the device state (`on`). The last entry is the pathname to the VSN catalog file in each of the library storage slots (`/etc/fs/samfs/hp30_cat`).

The two remaining lines define the drives inside the library. They are similar to the manually loaded drives defined in the previous section except that instead of a dash, they include the family set name of the library where they reside (`hp30`).

Configuring a Manually Loaded DLT Drive

When configuring DLT drives, make sure to add the DLT definitions to the `/kernel/drv/st.conf` file (see Step 4, “Add Tape Support to `st.conf`”). DLT drives are not part of the standard Solaris configuration.

The following lines from `/var/adm/messages` refer to the manual DLT drive:

```
Aug 23 11:52:54 baggins unix: samst4: Vendor/Product ID = DEC
DLT2000
Aug 23 11:52:54 baggins unix: samst4 at esp0: target 4 lun 0
Aug 23 11:52:54 baggins unix: samst4 is
/iommu@0,10000000/sbus@0,10001000/espdma@5,8400000/esp@5,880
0000/samst@4,0
```

Find the matching `/dev/samst` symbolic link:

```
lrwxrwxrwx   1 root     other        88 Aug 23 12:27 c0t4u0
->
/devices/iommu@0,10000000/sbus@0,10001000/espdma@5,8400000/e
sp@5,8800000/samst@4,0:a,raw
```

For tape devices you can leave the additional parameters field empty. The system finds the proper `/dev/samst/*` symbolic link using the Solaris st driver as described below.

Note: The additional parameters field is required if the equipment identifier field is not in the form /dev/rmt/* (the standard st device driver). In this case, the additional parameters field is the path to the "special file" (e.g., /dev/samst/cntnun).

For a tape device, there is another symbolic link located in `/dev/rmt`. This symbolic link is the name that the Solaris `st` driver (`see st(7)`) uses when referencing the device. There are many symbolic links in `/dev/rmt` that point to the hardware path. Each link has various combinations of the option letters `c`, `b` and `n`. When making the `mcf` entry, always use the `b` and `n` options prefixed with **`c`** if the drive supports compression. The symbolic link is:

```
lrwxrwxrwx   1 root     other         85 Aug 15 11:37
/dev/rmt/0cbn ->
../../devices/iommu@0,10000000/sbus@0,10001000/espdma@5,8400
000/esp@5,8800000/st@4,0:cbn
```

Using this information, construct the `/etc/fs/samfs/mcf` entry:

```
dev/rmt/0cbn  40  tp  -  on
```

The first entry on the line is the `st` driver name for the device (`/dev/rmt/0cbn`), followed by a unique ordinal (`40`), the equipment type (`tp` for a generic tape), a dash indicating a family set name is not associated with the manually-mounted device, and the device state (`on`).

Configuring a Robotically Loaded DLT Drive

The last piece of equipment left to define is the DLT 2700 library. This device occupies just one SCSI target, but has two logical unit numbers (LUNS) at that target. LUN #0 is the drive, while LUN #1 is the robot. Locate the `/var/adm/messages` entries for this device:

```
Aug 23 11:52:56 baggins unix: samst17: Vendor/Product ID = DEC
DLT2700
Aug 23 11:52:56 baggins unix: samst17 at QLGC,isp0: target 3
lun 0
Aug 23 11:52:56 baggins unix: samst17 is
/iommu@0,10000000/sbus@0,10001000/QLGC,isp@1,10000/samst@3,0
Aug 23 11:52:56 baggins unix: samst24: Vendor/Product ID = DEC
TZ Media Changer
Aug 23 11:52:56 baggins unix: samst24 at QLGC,isp0: target 3
lun 1
Aug 23 11:52:56 baggins unix: samst24 is
/iommu@0,10000000/sbus@0,10001000/QLGC,isp@1,10000/samst@3,1
```

Find the `/dev/samst` symbolic links which point to these hardware paths:

```
lrwxrwxrwx   1 root     other        74 Aug 23 12:27 c1t3u0
->
/devices/iommu@0,10000000/sbus@0,10001000/QLGC,isp@1,10000/s
amst@3,0:a,raw
lrwxrwxrwx   1 root     other        74 Aug 23 12:27 c1t3u1
->
/devices/iommu@0,10000000/sbus@0,10001000/QLGC,isp@1,10000/s
amst@3,1:a,raw
```

Since a tape device is involved, find a symbolic link in `/dev/rmt` which points to the device. Note that the robot doesn't have this additional link:

```
lrwxrwxrwx   1 root     root          71 Aug 23 11:52
/dev/rmt/2cbn ->
../../devices/iommu@0,10000000/sbus@0,10001000/QLGC,is
p@1,10000/st@3,0:cbn
```

Again, there are multiple symbolic links in the directory, which point to the same hardware path; choose the one with the `cbn` suffix because the drive supports compression. Had it not, you would have chosen the symbolic link whose name ended with `bn`.

Make the following entries in `/etc/fs/samfs/mcf`:

```
/dev/samst/c1t3u1  50  rb  ml50  on
/etc/fs/samfs/ml50_cat
/dev/rmt/2cbn      51  tp  ml50  on
```

The first line defines the robot, displaying the `/dev/samst` name (`/dev/samst/c1t3u1`), a unique ordinal (`50`), the equipment type (`rb`, for the generic robot equipment type), a family set name for the robot and the drive (`ml50`), the device state (`on`), and, as for all libraries, the name of a file in which to store this library's VSN catalog (`/etc/fs/samfs/ml50_cat`).

The second line defines the DLT tape drive inside the library. These entries refer to the equipment identifier for this tape device (`/dev/rmt/2cbn`), the ordinal for the device (`51`), the equipment type (`tp`), the family set name (`ml50`), and the device state (`on`).

Again, when configuring DLT drives, make sure that you add the DLT definitions to the `/kernel/drv/st.conf` file, as shown in Step 4, "Add Tape Support to `st.conf`". DLT drives are not part of the standard Solaris configuration.

Completed `/etc/fs/samfs/mcf`

Figure 2-2 shows the complete `/etc/fs/samfs/mcf` for the configuration example:

```
samfs1               1   ms   samfs1
/dev/dsk/c1t0d0s0   11   md   samfs1    on   /dev/rdsk/c1t0d0s0
/dev/dsk/c1t1d0s5   12   md   samfs1    on   /dev/rdsk/c1t1d0s5
samfs2               2   ms   samfs2
/dev/dsk/c1t1d0s0   15   md   samfs2    on   /dev/rdsk/c1t1d0s0
/dev/dsk/c1t0d0s1   16   md   samfs2    on   /dev/rdsk/c1t0d0s1
/dev/samst/c0t2u0   20   od   -         on
/dev/samst/c1t2u0   30   rb   hp30      on   /etc/fs/samfs/hp30_cat
/dev/samst/c1t5u0   31   od   hp30      on
/dev/samst/c1t6u0   32   od   hp30      on
/dev/rmt/0cbn       40   tp   -         on
/dev/samst/c1t3u1   50   rb   ml50      on   /etc/fs/samfs/ml50_cat
/dev/rmt/2cbn       51   tp   ml50      on
```

Figure 2-2 Complete `/etc/fs/samfs/mcf` Example

Step 10: Set up ASM Default Values

The `/opt/LSCsamfs/examples/defaults.conf` file contains default settings for certain default values in ASM. Read the `defaults.conf(4)` manual page and examine this file to determine which, if any, of the defaults should be enabled. Copy the example `defaults.conf` to `/etc/fs/samfs/defaults.conf`, then edit this file, removing comments from the entries to be enabled.

Step 11: Create a VSN catalog

For most SCSI-attached libraries, A VSN catalog is created automatically when ASM is initialized. The additional parameter field for the robot entry in the mcf file gives the pathname to the catalog.

However, for a network-attached robot, you must create a catalog at this time. See Chapter 4, “Managing Removable Media Libraries”, for instructions on how to build a VSN catalog.

Step 12: Label Tapes or Optical Disks

Sites that use manually loaded devices or libraries without a bar code reader must prepare media to be used.

To prepare removable media, complete the following step.

- Using `tplabel (1M)` or `odlabel (1M),` create a label on the removable media that SAM-FS uses (see examples).

  ```
  # odlabel –vsn OPTIC01 –new /dev/samst/c0t5u0
  # tplabel –vsn TAPE01 -new /dev/samst/c0t4u0
  ```

  The media is now ready to be used.

Step 13: Configure the Archiver

The archiver archives all files under all ASM mount points. Note that the administrator is not required to take action. The archiver archives to all VSNs in all configured robots. If your site has additional requirements, you need to set up an archiver command file. See the discussion on automatic storage management in Chapter 6, "Archiver Operations", and the `archiver.cmd(4)` manual page for additional information.

Step 14: Create the Mount Point and Update /etc/vfstab

This document assumes */sam/* is the mount point of the *samfs1* file system. You can select a different name and substitute it for */sam/*, if you want.

1. Create the mount point:

   ```
   server# mkdir /sam/
   ```

2. Next, if you want, change the permissions, owner, or group owner of the root directory of the file system

   ```
   server# chmod 755 /sam
   server# chown root /sam
   server# chgrp other /sam
   ```

3. Make the `/etc/vfstab` entry for each ASM file system. An example entry follows:

```
samfs1  -  /sam  samfs  -  yes  high=80,low=60
```

The first field (device to mount) specifies the name of the ASM file system to mount. The second file (device to fsck) contains a dash indicating there are no options: do not fsck an ASM file system. The third field (mount point) is the default mount point. The fourth field (FS type) must be `samfs`. The fifth field (fsck pass) is unused and contains a dash indicating there are no options.

The sixth field (mount at boot) specifies `delay`, a special flag which is interpreted by a script which is installed into the `/etc/rc3.d` directory by the `pkgadd` command. This script, which is run when the system enters init state 3, scans the `/etc/vfstab` file and mounts `samfs` file systems which are flagged `yes`. See the `mount_samfs(1M)` manual page for the format of these entries. Specifying `no` indicates that you do not want to automatically mount the file system.

Finally, the seventh field (mount options) is a list of comma-separated options (with no spaces) that are used in mounting the file system. See `mount_samfs(1M)` for a list of the available mount options.

Note: If you configured multiple mount points, repeat these steps for each mount point, using a different mount point (*/sam*) and family set name (*samfs1*) each time.

Step 15: Make the File System

1. Using `sammkfs`, create a file system for each family set that has been defined.

```
server# /opt/LSCsamfs/sbin/sammkfs samfs1
```

An output sample from sammkfs is included:

```
server# sammkfs samfs1
total data kilobytes       = 31842048
total data kilobytes free  = 31841680
```

Caution: Running `sammkfs` creates a new file system, removing all data currently contained in the partition associated with the file system in the `/etc/fs/samfs/mcf`.

Step 16: Startup and Shutdown ASM

The `mount_samfs` command to mount an asm file system starts `sam-init` if the system is at run-level 3 or above.

`/etc/rc3.d/S95samd` starts up `sam-init` if the `/etc/fs/samfs/mcf` exists and `sam-init` is not already running.

Change the `/etc/vfstab` mount at boot parameter to "yes". This indicates that samfs file systems will be mounted by `/etc/rc2.d/S01MOUNTFSYS`. It is possible to stop `sam-init` and leave the samfs file systems mounted. When `sam-init` is restarted, pending stages are reissued and archiving is resumed.

Manual Startup/Shutdown

1. Enter the following information to perform manual startup:
   ```
   server# samd start
   server# mount /asm
   ```

2. Enter the following information to perform manual shutdown:
   ```
   server# samd stop
   server# umount /asm
   ```

Automatic Startup/Shutdown

1. Enter the following information to perform an automated startup:
   Edit `/etc/vfstab` to ensure that `yes` is enabled.
   Boot system. Go to run level 3 (multi-user mode).

2. If `sam-init` is not running, stages are not queued and files are not archived until `sam-init` is started.

Step 17: Share the File System with Client Machines

The Solaris `share(1M)` command must be run to make the file system available for mounting by remote systems. Share commands are typically placed in the `/etc/dfs/dfstab` file and are executed automatically by Solaris when entering `init` state 3.

For example, on the server, enter the line:

```
server# share -F nfs -o rw=client1:client2 -d "ASM" /sam
```

Note: If you make the above entry in `/etc/dfs/dfstab`, Solaris shares the file system after the next system reboot. If you want to share the file system immediately, you must type the `share` command at a root shell prompt. If there are no shared file systems when Solaris boots, the NFS server is not started. You must reboot after adding the first share entry to this file.

The following NFS mount parameters can impact the performance of a NFS mounted SAM-FS file system. You can set the following parameters in the `/etc/vfstab` file (see `mount_nfs(1M)`).

```
timeo = n
```

This value sets the NFS timeout to *n* tenths of a second. The default is 11 tenths of a second. LSC recommends setting this number to a larger value such as 1000 or even 10000.

```
rsize = n
```

This value sets the read buffer size to *n* bytes. In NFS 2, change the default value (8192) to 32768. In NFS 3, retain the default value of 32768.

```
wsize = n
```

This value sets the write buffer size to *n* bytes. In NFS 2, change the default value (8192) to 32768. In NFS 3, retain the default value of 32768.

Step 18: Mount the File System on the Client Machines

On the client systems, mount the server's `/sam` file system at a convenient mount point. In this example, we mount *server*`:/sam` on `/sam` and enter the following information in `/etc/vfstab`:

```
server:/sam - /sam    nfs - no hard,intr,timeo=60
```

Next, on the command line, perform the `mount`:

```
client# mount /sam
```

The automounter can also do this, if the site prefers. Follow your site procedures for adding *server:/sam* to your automounter maps.

Note: It is strongly recommended that clients mount the file system with the `hard` option. At times, there may be a significant delay in ASM's response to client requests. This can occur when a requested file resides on a piece of media that must be loaded into a DLT tape drive. If the `hard` option is not given, the client may return an error instead of retrying the operation until it completes.

If you must use the `soft` option, make sure you set the value of `retrans` to a large number such as 120 (the default is 5). This sets the number of NFS retransmissions.

Step 19: Establish Periodic samdumps

The server should periodically create a control structure dump using `samdump(1M)`. This dump does not include the data stored in your file system, but it does include information necessary to quickly locate the data on your removable media devices. This information is necessary to recover from a cache disk failure. Use `samrestore` to restore the control structure dump after initializing the file system if such a failure occurs. Make an entry in root's `crontab` file so the `cron` daemon runs `samfsdump` periodically:

```
10 0 * * * (find /csd.directory -type f -mtime +3 \
print| xargs -l1 rm -f); cd sam1; \
/opt/LSCsamfs/sbin/samfsdump -f \
csd.directory/sam1/‘date +\%y\%m\%d‘
```

This example crontab entry uses an ASM file system named `sam1`. Replace `/csd.directory` with an existing directory of your choice. This entry causes the commands to be executed each day at midnight. First, the old dumps are renamed and a new dump is created in `/csd.directory/sam1/`*yymmdd*. `cron` mails the output of `samfsdump` to root.

If you have multiple ASM file systems, make similar entries for each. Make sure you save each dump in a separate file. See the `samdump(1M)` manual page and Chapter 9, “Dumping and Restoring ASM Control Structures”.

Step 20: Establish Periodic Backups of .inodes

As an alternative to `samdump(1M)`, you can periodically save inode information, which is stored in file `.inodes` under the mount point. The `.inodes` file is used to recover a file system after a cache disk failure. Save the file using the following command:

```
server# dd if=/sam/.inodes of=/home/samfs/inodes
bs=128k
```

After a disk failure, the backup `.inodes` file can be input to `sammkfs(1M)`. `sammkfs` constructs the file system and restores the inodes. All the files, directories, symbolic links, and removable media files are offline. You can back up the `.inodes` file more frequently since it takes less time than running `samfsdump`. Specifying the block size in integer multiples of 16k increases the performance of the `dd` copy (e.g., `bs=128k` or `bs=512k`).

Software Upgrade Procedure

This section describes upgrading a server to a new release of the ASM software. All steps in this section must be performed as superuser (root).

Step 1: Back Up Each ASM File System

If you do not have current backup files for your ASM file systems, create them now using `samdump` and by copying the inodes file using `dd`. See Step 19, “Establish Periodic `samdumps`” and Step 20, “ Establish Periodic Backups of `.inodes`” in the previous section on ASM installation and configuration procedures.

Step 2: Mount the Installation CD-ROM

Insert and mount the ASM installation CD-ROM as described in Step 2, “Mount the Installation CD-ROM” in the previous sub-section on ASM installation and configuration procedures. Note that you should follow the installation instructions accompanying the CD-ROM. These instructions provide up-to-date details on adding the ASM release files to the server.

Running the Solaris Volume Manager, enter the following:

```
Insert CD-ROM
    server# cd /cdrom/cdrom0
```

Step 3: Stop sam-init

To stop `sam-init`, type:

```
server# /opt/LSCsamfs/sbin/samd stop
```

Step 4: Unmount the File Systems

Using the Solaris `umount` command, unmount each ASM file system. If you encounter difficulty unmounting a file system, it may be because you or another user is using files or has changed to directories in the file system. If you cannot identify and correct this situation, you may need to reboot with the mount at boot field in `/etc/vfstab` changed from `delay` to `no`. This inhibits the file systems from being mounted at reboot time. You can use the "`fuser` *`samfs1`*" command to show the process IDs of processes using the *`samfs1`* file system.

Step 5: Remove Existing ASM Software

Use `pkgrm` to remove the existing ASM software. You must remove all existing ASM packages before installing the new packages. If you are using any of the optional StorageTek packages as described at the beginning of this chapter, you should make sure that you remove these packages prior to the main StorageTek asm package. Use `pkginfo(1)` if you are unsure as to which packages are currently installed.

The following example removes all of the StorageTek packages:

```
server# pkgrm LSCibm
server# pkgrm LSCstk
server# pkgrm LSCdst
server# pkgrm LSCgui
server# pkgrm LSCjre
server# pkgrm LSCdoc
server# pkgrm LSCsamfs
```

Step 6: Unload ASM Modules

To avoid a reboot, you need to unload the ASM modules. The following example checks to see if the modules are loaded, and in this case, unloads them:

```
server# modinfo | grep sam
 91 f5ef4000 3904c 17 1  samfs (Storage and Archiving
Mgmt FS)
 92 f5c3c800 3cc  180 1  samsys (SAMFS system)
 93 f6015000 109   17 1  samst (SAMFS optics/robots SCSI
target)
server# modunload -i 91
server# modunload -i 92
server# modunload -i 93
```

If you cannot unload the modules, you need to reboot the system as described in Step 10, "Reboot the System".

Step 7: Add the Packages

ASM uses the Solaris packaging utilities for adding and deleting software. As such, you must be logged in as superuser (root) to make changes to software packages. `pkgadd(1M)` prompts you to confirm various actions necessary to upgrade the StorageTek packages.

The ASM packages and all optional products reside in the `/cdrom/cdrom0` directory. To satisfy product dependencies, you must upgrade the `sampkg` first. Run the `pkgadd` command to upgrade all packages, answering "yes" to each question:

```
server# pkgadd -d sampkg(must be first)
server# pkgadd -d samstk(optional)
server# pkgadd -d samdst(optional)
server# pkgadd -d samibm(optional)
```

An optional set of graphical user interfaces (GUIs) are available with ASM. These GUIs will eventually replace our current tools under `samtool(1M)`. The new GUIs require the installation of a Java runtime environment. Add these packages as follows:

```
server# pkgadd -d samgui(optional)
server# pkgadd -d samjre(optional)
```

This manual is available in both pdf and postscript formats. Add this package as follows:

```
server# pkgadd -d samdoc(optional)
```

Step 8: Restore File Changes (inquiry.conf and samst.conf)

StorageTek does not guarantee correct operation with peripherals other than those included in the `/etc/fs/samfs/inquiry.conf` file supplied with the release. Therefore, each release overwrites the existing file. When adding the new package, ASM saves a copy of this file in `/etc/fs/samfs/inquiry.conf.`*`mmddyy`*. Compare the old file with the new file and merge any changes that might be used with your configuration.

If you have modified `/kernel/drv/samsf.conf`, you need to merge any changes to it as well.

The original `/kernel/drv/samst.conf` file is backed up to an alternate file during the upgrade before it is replaced with the latest version during the installation of the LSCsamfs package. The original `/kernel/drv/samst.conf` file is written to `/kernel/drv/samst.conf.mddyy` during installation.

If necessary, you should manually transfer any local modifications to the new `samst.conf` file from the backup version.

Step 9: Update the License Keys

It may be necessary to update the license keys. If you are upgrading from a ASM release prior to 3.3.0 you need to place a new license key in `/etc/fs/samfs/LICENSE.3.3`. For more information see Step 7, "License ASM" in the ASM installation and configuration procedures section of this chapter.

Step 10: Reboot the System

If you were unable to unload the ASM modules described in Step 6, “Unload ASM Modules” in this sub-section, you must reboot the system at this time. Otherwise, continue with Step 11.

Step 11: Mount the File System(s)

Mount the file systems and continue operation with the upgraded ASM software.

Step 12: Relink API Applications (optional)

If you are running applications that use the ASM application programmer interface (API) and you are using static linking, you should relink these applications at this time.

Chapter 3. ASM Basic Operations

Introduction

This chapter describes the basic operations for direct SCSI-attached removable media libraries and manually mounted devices. The following topics are presented:

- Basic operations - Common to all media libraries.
- Media library operations - For systems that use a mailbox.
- Media library operations - For systems that do not use a mailbox.
- Manually-mounted device operations - For stand-alone tape and magneto-optical drives

If you have a library that is network-attached versus direct SCSI-attached, see the respective manual pages for `stk`(7), `grauaci`(7), `ibm3494`(7), and fujitsulmf(7).

Basic Operations

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

- How to start and stop ASM
- How to turn a library “on” and “off”
- How to audit a volume
- How to audit a library
- How to mount (load) a volume
- How to unload a volume from a drive
- How to label media
- How to clean a tape drive
- Clearing media errors

How to Start ASM

`SAM-init` is started automatically by mounting an ASM file system.

You can also start `SAM-init` without mounting any file systems by entering the following:

```
# /opt/sam/sbin/samd start
```

If ASM is already running, the following message is displayed:

```
SAM-init daemon already running
```

How to Stop ASM

It is possible to stop `sam-init` and leave the ASM file systems mounted. When `sam-init` is restarted, pending stages are reissued and archiving is resumed.

To stop `sam-init`, type:

```
# /opt/sam/sbin/samd stop
```

If `SAM-init` is not running, stages will be queued and files will not be archived until `sam-init` is started.

How to Turn a Library "On"

When a library is in the "on" state, it is under ASM control and may proceed with general operations. The `samu(1M)` "c" display and the `robottool(1M)` robots panel display the state of the library.

To turn a library to "on", enter one of the following:

- From `samu(1M)`, enter the following command where *eq* is the ordinal of the library:

  :on *eq*

- From `robottool(1M)`, select the library in the robots panel. Then select the "Change State" button, pull down the menu, and select "On".

How to Turn a Library "Off"

Placing a library in the "off" state stops I/O operations and removes the library from ASM control. No automatic robotic movement of media occurs. Note that the drives in the robot remain in the "on" state. You might turn a library off when:

- You want to stop ASM operations for this library only.
- You suspect that the library or a drive is not functioning properly and you want an engineer to run diagnostics.
- You want to power down the library.

To turn a library to "off", enter one of the following:

- From `samu(1M)`, enter the following command where *eq* is the ordinal of the library:

  :off *eq*

- From `robottool(1M)`, select the library in the robots panel. Then select the “Change State” button, pull down the menu, and select “Off”.

How to Load a Volume into a Drive

Loading a volume in a drive occurs automatically when the volume is needed for archiving or staging. Loading refers to moving a tape or optical from a storage slot to a drive and making it ready. When you manually load a volume, it is loaded in the next available drive in the library; you cannot explicitly load a volume to a drive. (If you wish to make a drive unavailable for this purpose, use the `samu(1M)` “:unavail” command or change the state of the device using `devicetool(1M)`.

To manually load a volume, enter one of the following:

1. From `samu(1M)`, load the volume by the slot number (*`slot`*) or VSN (*`vsn`*). Enter one of the following commands, where *`eq`* is the ordinal of the library:

   :mtslot *eq slot*

   :mtvsn *vsn*

2. From `robottool(1M)`, select the VSN from the VSN catalog panel. Select the “Mount” button. The robot loads the medium in a drive.

How to Unload a Volume from a Drive

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

To manually unload a drive, enter one of the following:

- From `samu(1M)`, unload the drive by entering the following command, where *`eq`* is the equipment number of the drive:

  `:unload` ***`eq`***

- From `devicetool(1M)`, select the drive in which the VSN is present. Then select the “Unload” button. The robot unloads the medium from the drive and places it in its storage slot.

How to Label Media

If you are not using an automated library with a barcode reader, you must label the media before using ASM.

If your automated library uses barcodes, make sure that the following line is included in the `/etc/fs/samfs/defaults.conf` file. This line ensures that the six low-order digits become the VSN for the medium.

```
labels = barcodes_low
```

If you want the six high-order digits to become the VSN label, include the following line in the `/etc/fs/samfs/defaults.conf file`:

```
labels = barcodes
```

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

Caution: Labeling and relabeling the media will destroy all data on the media. Make sure this is your intent before labeling or relabeling media.

Select one of the following methods to label a new tape:

- Enter the following information:

  ```
  # tplabel  -new -vsn new_vsn -slot slot eq
  ```

  where ***new_vsn*** is the VSN for the new tape, ***slot*** is the slot number in which the tape is located, and ***eq*** is the equipment number for the robot as defined in `mcf`(4).

OR

- Use `robottool`(1M) to select a medium. Click the Label button and enter the information for the new VSN as prompted.

  The tape is mounted and positioned, and the new tape label is written.

Select one of the following methods to relabel a tape:

- Enter the following information:

  ```
  # tplabel  -old old_vsn -vsn new_vsn -slot slot eq
  ```

  where ***old_vsn*** is the volume serial name of the old tape, ***new_vsn*** is the VSN for the new tape, ***slot*** is the slot number in which the tape is located, and ***eq*** is the equipment number for the robot as defined in `mcf`(4).

OR

- Use `robottool`(1M) to select a medium. Click the Label button and enter the information for the old VSN and new VSN as prompted.

  The tape is mounted and positioned, and the tape label is written.

Select one of the following methods to label a new optical disk:

- Enter the following information:

  # **odlabel -new -vsn** ***new_vsn*** **-slot** ***slot eq***

  where ***new_vsn*** is the VSN for the new optical disk, ***slot*** is the slot number in which the optical disk is located, and ***eq*** is the equipment number for the robot as defined in `mcf`(4).

OR

- Use `robottool`(1M) to select an optical disk. Click the Label button and enter the information for the new VSN as prompted.

  The optical disk is mounted and positioned, and the new optical disk label is written.

Select one of the following methods to relabel an optical disk:

- Enter the following information:

  # **odlabel -old** ***old_vsn*** **-vsn** ***new_vsn*** **-slot** ***slot eq***

  where ***old_vsn*** is the volume serial name of the old optical disk, ***new_vsn*** is the VSN for the new optical disk, ***slot*** is the slot number in which the optical disk is located, and ***eq*** is the equipment number for the robot as defined in `mcf`(4).

OR

- Use `robottool`(1M) to select an optical disk. Click the Label button and enter the information for the old VSN and the new VSN as prompted.

  The optical disk is mounted and positioned, and the optical disk label is written.

How to Audit a Volume

Occasionally the reported capacity of a tape or optical medium may need to be updated in the catalog. Performing an audit loads the medium, reads the VSN, and updates the catalog entry for the slot.

To perform an audit, on a volume enter one of the following:

- From the following command line, where *eq* is the equipment ordinal of the library:

```
# auditslot -e  eq slotnumber
```

- From `robottool(1M)` select the VSN from the VSN catalog panel. Select the “Audit” button. The robot mounts the medium in to a drive and updates the catalog.

How to Audit a Library

A full audit mounts each piece of media into a drive, reads the label, and updates the library catalog. A library should be audited in the following situations:

- After moving media in the library without using ASM 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 media in a library that has no mailbox. (See “Media Library Operations - Systems Without a Mailbox” in this chapter.)

To perform a full audit on a media library enter one of the following:

- From `samu(1M)` enter the following command, where *eq* is the equipment ordinal of the library:

  ```
  :audit eq
  ```

- From `robottool(1M)` select the library in the robots panel. Select the “Full Audit” button. The following message appears: “This command will take a long time. Do you wish to Continue or Cancel?” Select “Continue” to proceed with the audit.

How to Clean a Tape Drive

ASM supports the use of cleaning tapes, if cleaning tapes are supported by the hardware and your media library has `labels = barcodes` enabled. If a tape drive requests to be cleaned, ASM automatically mounts a cleaning tape.

Cleaning tapes must have a VSN of “CLEAN” or a VSN starting with the letters "CLN" in the barcode label. Multiple cleaning tapes are allowed in a system.

Note: Certain drive errors can result in cleaning media being mounted repeatedly until all cleaning cycles are consumed. Sites can prevent this situation by limiting the number of cleaning cycles on cleaning media with the `chmed(1M)` command.

When automatic cleaning is not available and the system uses barcodes, you can invoke the `cleandrive(1M)` command to manually request that a drive be cleaned:

```
# cleandrive eq
```

where *eq* is the tape device equipment ordinal to be loaded with a cleaning cartridge.

Cleaning tapes are only useful for a limited number of cleaning cycles. The number of remaining cycles can be viewed in the `robottool(1M)` VSN catalog display under the "count" field, or with `samu(1M)` in the “`v`” display. ASM tracks the number of cleaning cycles used for each cleaning tape and ejects the tape when the remaining cycles equals zero. 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 library have a count of zero, the drive is set to off and a message is issued in the ASM log. A cleaning tape with a count of zero can be reset using the `chmed(1M)` command as follows:

```
# chmed -count count media_type vsn
```

count is the number of cleaning cycles you want to reset. *media_type* specifies the media type. See `media(5)` for the complete list of media types. *vsn* specifies the volume serial name of the media.

Automatic cleaning by ASM is not supported on network-attached libraries. You should use the vendor's library manager software for automatic cleaning.

Clearing Media Errors

When a hardware or software error is encountered on a piece of media, ASM sets the "MEDIA ERROR" flag in the VSN catalog. On any given piece of media which gets a media error, the `chmed(1M)` command can be used to clear the error and you can attempt to use the piece of media. The "MEDIA ERROR" flag is displayed in the `samu(1M)` "v" display, in the `robottool(1M)` VSN catalog panel, and in the `libmgr(1M)` VSN display.

To clear the media error flag using `chmed`, enter the following:

chmed -E *media_type vsn*

`media_type` specifies the media type. See `media(5)` for the complete list of media types. *`vsn`* specifies the volume serial name of the media.

Media Library Operations - Systems with a Mailbox

This section describes how to import and export media for tape and optical media libraries that use a mailbox.

A *mailbox* is an area where media is added and removed from the library. The *import* calls ASM to move a cartridge from the mailbox to a storage slot. The *export* command calls ASM to move the cartridge from a storage slot to the mailbox. If your system does not have a mailbox, the operations described in this section do not apply to your system. See the next section, "Media Library Operations - No Mailbox Systems".

Some media libraries only import and export one piece of media at a time. A *mail slot* handles a single piece of media and is used to import to and export from the library.

Libraries that use a mailbox include the StorageTek 9714, StorageTek 9710, and ADIC Scalar series media libraries. Libraries that use a mail slot include the HP SureStore st series.

How to Import Tapes Using a Mailbox

To import tapes into a library that uses a mailbox, follow these steps:

1. Open the mailbox using the manufacturer's suggested operation (usually a button near the mailbox). Sometimes the mailbox is a one-slot mailbox referred to as a *mail slot*.

2. Manually place the media into the mailbox or mail slot.

3. Close the mailbox.

4. Import the media by entering one of the following commands. ASM moves the media from the mailbox to a storage slot and updates the library catalog for each piece of media.

   From the command line enter:

   ```
   # import eq
   ```

   where *eq* is the equipment ordinal of the library.

5. From `robottool(1M)`, select the library in the robots panel. Then select the “Import” button.

6. From `samu(1M)`, enter the following command, where *eq* is the the equipment ordinal of the library:

   ```
   :import eq
   ```

How to Export Tapes Using a Mailbox

1. To export (remove) tapes from a library that uses a mailbox, select one of the following methods. The media is moved from a storage slot to the mailbox or mail slot.

   From the command line:

   Enter one of the following export commands where *slot* is the number of the slot in which the media is stored, *vsn* is the volume serial name of the media, and *eq* is the ordinal of the library:

   ```
   # export -s slot eq
   ```

   OR

   ```
   # export -v vsn eq
   ```

   From `robottool(1M)`:

   Select the slot or VSN in the VSN catalog panel, and then select the “Export” button.

   From `samu(1M)`:

   Enter one of the following commands, where *eq* is the equipment ordinal of the library and *slot* is the slot number in which the tape resides:

   ```
   :export eq slot
   ```

2. Open the mailbox or mail slot using the manufacturer’s suggested operation (usually a button near the mailbox).

Media Library Operations - Systems Without a Mailbox

This section describes operations for media libraries that do not use a mailbox for media movement. To move media into and out of the library, idle the library, open the system door, remove or add media, and close the door. If the library does not use barcode media, an audit must be performed.

How to Add/Remove/Move Tapes (No Mailbox)

Libraries that do not have a mailbox or mail slot do not have the capability of importing or exporting tapes. Therefore, after every movement of tape within these libraries, the library catalog needs to be updated either automatically by scanning the barcodes or, if barcodes are unavailable, by auditing the library.

To add, remove, move tapes in a library that has no mailbox, follow this sequence:

1. Ensure that no archive or stage processes are active by idling all of the drives in the library. Enter one of the following:

   From `samu(1M)` enter the following, where *`eq`* is the ordinal the drive:

   ```
   :idle eq
   ```

   From `devicetool(1M)` select the drive in the devices panel. Select the "Change State" button, pull down the menu, and select "Idle".

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

If your library does not use barcodes:

2. Move the media catalog entries into the historian using the `unload(1M)` command or unload on `robottool(1M)`. Enter one of the following:

   From `samu(1M`, unload the catalog by entering the following command, where *`eq`* is the ordinal of the library:

   ```
   :unload eq
   ```

   From `robottool(1M)` select the library, then select the "Unload" button. The robot catalog is immediately emptied; these entries are now visible if you select the historian (hy) catalog.

3. Unlock and open the door to the library.

4. Now you can add media, remove media, or move media within the slots of the library.

5. Close the door to the library and lock it.

If your library uses barcodes:

The robot re-initializes and scans the media into the library. ASM updates the catalog, which should now contain the tapes' VSNs. The library state is "on".

If your library does not use barcodes:

6. Complete a full audit of the library to update the catalog to reflect the repositioned media. To perform a full audit on a media library enter one of the following:

   From `samu(1M)` enter the following command, where *`eq`* is the ordinal of the library:

   ```
   :audit eq
   ```

   From `robottool(1M)` select the library in the robots panel. Select the "Full Audit" button. The following message appears: "This command will take a long time. Do you wish to Continue or Cancel?" Select "Continue" to proceed with the audit. The library will load each piece of media, read the label, and add the media entry in to the library catalog.

Manually Mounted Drive Operations

This section describes how to load and unload a medium with a manually mounted (or stand-alone) drive.

Manually mounted drives use the historian's catalog as the working catalog. Select the historian (hy) entry to display the catalog entries in the `robottool(1M)` by. Use `devicetool(1M)` to load and unload the media in manually mounted devices.

How to Load a Medium

To load a tape cartridge or magneto-optical platter into a manually mounted device, place the cartridge in the drive according to the manufacturer's instructions. ASM recognizes that a piece of media is loaded, and it reads the label and updates the catalog. No further action is necessary.

How to Unload a Medium

1. Select one of the following methods to idle the drive. This ensures that no archive or stage processes are active.

   From `samu(1M)` enter the following, where *`eq`* is the ordinal of the drive:

   ```
   :idle eq
   ```

   OR

   From `devicetool(1M)` select the drive in the devices panel, select the "Change State" button, pull down the menu, and select "Idle".

   – The drive switches from "idle" to "off" when all I/O activity is completed.

2. From `samu(1M)`, unload the drive by entering the following command, where *`eq`* is the ordinal of the drive:

   ```
   :unload eq
   ```

   From `devicetool(1M)` select the drive. Then select the “Unload” button.

   The tape rewinds and is ready to be removed. An optical platter ejects automatically. See the manufacturer’s instructions for removing the specific media.

Chapter 4. Managing Removable Media Libraries

Introduction

This chapter describes the theory and operations of managing removable media libraries. The following topics are presented:

- Library hardware and software components
- Configuring libraries
- Library operations
- Managing exported volumes using the historian

Media Library Hardware and Software

A media library is a robotically-controlled device designed to load and unload removable media under programmed control without operator intervention. A library consists of three parts: a *transport* that moves media to and from *storage* and *drives*. Libraries may also be known as media changers, jukeboxes, or simply robots. Media is imported to and exported from the library, loaded and unloaded automatically or manually, and archiving and staging proceeds using a site-defined scheme for allocating the number of drives to use.

A *SCSI-attached library* is connected directly to a server using SCSI interfaces, such as the HP optical libraries. These libraries are controlled directly by ASM using the SCSI standard for media changers.

A *network-controlled library* such as some of the StorageTek (STK), EMASS/Grau, or IBM libraries, are controlled using a software package supplied by the vendor. In these cases, ASM interfaces with the vendor software using an LSC, Inc. media changer daemon specifically designed for the library. The media changer daemons are shown as follows:

Media Changer Daemon	Description
`robots`	Monitors the execution of robot control daemons `robots` is started automatically by `sam-init`.
`generic`	Controls libraries that conform to SCSI II standards for media changers and the network-controlled EMASS/Grau and Fujitsu LT300 libraries.
`stk`	Controls the StorageTek (STK) media changers through the ACSAPI interface. (Optional package available from LSC, Inc.)
`ibm3494`	Controls the IBM 3494 tape libraries through the lmcpd interface. (Optional package available from LSC, Inc.)

The IBM 3494 and StorageTek network-attached libraries and the Ampex libraries require an additional software package to run ASM. These optional packages are listed in Chapter 2, "ASM Installation and Configuration". They also require vendor-supplied drivers.

How to Configure Media Libraries

Libraries are configured in the `/etc/fs/samfs/mcf` file. The additional parameters field in the `mcf` file contains the pathname of a *catalog file*, which contains information about the contents of each of the media storage slots in the library. Some libraries, such as the IBM 3494, StorageTek, and Grau, 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.

ASM configuration of libraries should not be attempted 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 master configuration file (`mcf`) is written as described in Chapter 2, "ASM Installation and Configuration". Drives are associated with a library using a family set name, each device is given a unique equipment ordinal, and so on. However, two important fields must be supplied within the entry for the library: the equipment identifier field, and the additional parameters field.

For SCSI-attached libraries, the equipment identifier field is the `/dev/samst` entry for the library itself. See the configuration example for a magneto-optical library in Step 9, "Configure ASM", in Chapter 2, "ASM Installation and Configuration". For network-controlled robots such as the STK, Grau, and Fujitsu, the equipment identifier field contains the full path name to the parameters file (described in the next subsection).

For both SCSI and network-controlled libraries, the additional parameters field is the full pathname to the library catalog.

The following `mcf` entries define an STK network-attached library. In this example, two drives are associated with an STK library.

# Equipment	eq	eq	family	dev	Additional
# Identifier	ord	ty	Set	st	Parameters
#					
/etc/fs/samfs/stk50	50	sk	Stk50	on	/etc/fs/samfs/stk50_cat
/dev/rmt/0cbn	51	sg	Stk50	on	
/dev/rmt/1cbn	52	sg	Stk50	on	

The parameters file

If the library is using the vendor's software package running an interface as previously described, you must set up an additional parameters file that defines to ASM the system characteristics of the library and its drives. The parameter file is different for each vendor's library; refer to the manual page for the appropriate vendor to review the available parameters. See the `stk(7)`, `ibm3494(7)`, `grauaci(7)`, and `fujitsulmf(7)` manual pages for complete descriptions of each system.

The following example parameters file (`/etc/fs/samfs/stk50`) is from an StorageTek library. In this example, two drives are described for the StorageTek library in accordance with the `stk(7)` manual page.

```
#
#  This is file:  /etc/fs/samfs/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 parameters you can set in the `/etc/fs/samfs/defaults.conf` file to control robot operations. See `defaults.conf(4)` for a complete description of each parameter.

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, as follows:

`labels = barcodes`	Use the first part of barcode as the label
`labels = barcodes_low`	Use the last part of barcode as the label
`labels = read`	Read the label from the tape

If `labels = barcodes` or `barcodes_low`, any tape mounted for a write operation that is write enabled, unlabeled, and has a readable barcode, will have a label written before the write is started.

Drive Timing Values

You can control the unload wait time and dismount time for devices by using *dev*`_delay` and *dev*`_unload`. Set the values for these parameters in accordance with your site's requirements. *dev* is the device type as specified in the `media(4)` manual page.

dev`_delay`

dev`_delay` indicates the dismount time in seconds for device type *dev*. After media is loaded on the device type, the indicated time must elapse before the media is dismounted and another media is mounted. The default for *dev***_delay** is 30 seconds.

dev`_unload`

dev`_unload` indicates the unload wait time in seconds for device type *dev*. This indicates time the media changer daemons wait after the device driver returns from a SCSI unload command. This gives the media changer time to perform tasks (e.g., ejecting the media, opening the door, etc.) before the daemon commands the changer to remove the media. Set this value to the longest time needed for the worst case media changer configured. See the example file that is supplied with the installation for default values (`/optLSCsamfs/examples/defaults.conf`). If a device is not included in the example file, it defaults to 0.

Example

The following example contains several parameters that can impact the configuration of a library. It includes several of the available keyword/value pairs. A sample is located in `/opt/LSCsamfs/examples/defaults.conf`.

See `defaults.conf(4)` for a complete listing of keywords.

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

Library Operations

There are several operations necessary to initialize and maintain your library under ASM. This subsection shows you how to build a catalog, and how to add and remove media from the library.

A catalog is a binary UFS-resident file, which contains information about each slot in a library. The information includes the VSN of the medium stored in the slot, the capacity and space remaining on that medium, and flags indicating read-only, write-protect, recycling, and other status for the medium. Therefore, the catalog is the central repository of all information that ASM needs to find media in the library.

ASM treats catalogs differently depending on how the library is attached. If the library is a SCSI-attached library, then the catalog is a one-to-one mapping between catalog entries and physical slots in the library. The first entry in the catalog is for the first slot in the library. When a medium is needed, ASM consults the catalog to determine which slot contains the VSN and issues a SCSI command to load that slot into a drive.

If the library is not a SCSI _attached_ library, then the catalog is not a direct mapping to the slots, but simply a list of the VSNs known to be present in the library. When a medium is requested, ASM sends to the vendor's software a request to load the VSN into a drive. The vendor's software handles locating the VSN's storage slot.

How to build a catalog

When the `robots` daemon starts, it checks to see that the catalog file for its library is present. If a library's catalog file is not present, then the direct SCSI-attached and IBM 3494 libraries perform a full audit of the library and build the catalog. For tape libraries, the audit notes the location and barcode (if available) of all media in the library. For optical libraries, the audit involves mounting all media and reading the labels. This information is stored in the newly-created catalog.

However, StorageTek, EMASS/Grau, and Fujitsu libraries cannot be audited in this way. For these libraries, the administrator must create an initial catalog. Then, using a series of `import` commands, the administrator instructs ASM to communicate with the library to fill in the catalog. The following example describes how to create such a catalog.

Assume a system with an StorageTek library. The `mcf` file entry for the robot appears as:

```
# Equipment              Eq    Eq   Family     Dev   Additional
# Identifier             Ord   Ty   Set        St    Parameters
#
/etc/fs/samfs/stk50      50    sk   sk50       on    /etc/fs/samfs/sk50_cat
```

First, build the empty catalog. The following example builds a catalog with 500 entries:

```
server# build_cat-s 500 /etc/fs/samfs/sk50_cat <
/dev/null
```

Then, bring up `sam-init` and issue `import` commands to load the catalog with the media already present in the library:

```
server# import -v volser1 50
server# import -v volser2 50
server# import -v volser3 50
```

This method is acceptable if there are not too many media in the library. For many volumes, an alternate method can be used. The `build_cat` command can be given a list of VSNs, which it uses to populate the initial catalog. See the `build_cat` manual page for more information.

dump_cat The `dump_cat(1M)` command dumps the media changer catalog file in text format. It has two options: "–o" and "–V".

-o Lists media that is no longer present in the catalog (i.e., the in-use flag is not set but there is an entry present).

–V The verbose option displays flags and label times as comments. It prints a line for each VSN, displaying the label time, last modification time, and mount time. It displays the flags in the same format as the `samu v` display.

The following samples display information using the two `dump_cat` options.

```
root# dump_cat -o -V ad30_cat
# audit_time Wed Dec 31 18:00:00 1969
# version 1  count 48 mediatype lt
# Slot  VSN  Barcode Type PTOC Access Capacity Space Status
Sector Label Time
#      ---status---  ---label time----  --last mod time--  -
---mount time---
#
  0  SLOT1 NO_BAR_CODE lt  0xaf09  149 9955720 3537680
0x68000000 131072 0x371ce29a
#      -il-o-------  04/20/99 15:24:58  05/12/99 03:39:46
05/12/99 03:22:47
  1  00081   0081 lt  0xf427   61 15166196 6216784 0x62000000
131072 0x37307c8a
#      -il---b-----  05/05/99 12:14:50  05/12/99 03:03:09
05/12/99 03:02:28
  2     ?  00044 lt  0x4959   10 15153864 9839740 0x5a000000
262144 0x33d649b5
#      -i-Eo-b-----  07/23/97 13:13:09  03/25/99 13:37:22
03/25/99 13:33:19
  3  00060   0060 lt  0xde64   37 9939712 2011560 0x7a000000
131072 0x370400df
#      -ilEo-b-----  04/01/99 17:27:27  04/23/99 01:55:55
04/23/99 00:54:34
# 8  00077 NO_BAR_CODE lt  0xf0ce   55 15228912 6366612
0x38000000 131072 0x3717d1b6
#      --lEo-------  04/16/99 19:11:34  04/30/99 19:41:24
04/30/99 18:39:25
 9 CLN033 NO_BAR_CODE lt     0  18     0     0 0x68000000
0    0
#      -il-o-------  12/31/69 18:00:00  12/31/69 18:00:00
09/01/98 14:39:57
```

How to import and export media

The physical addition (import) and removal (export) of media from a library can perform several functions (e.g., replace volumes, ship disaster recovery tapes off site). The ASM robot manager accomplishes these tasks using the `import(1M)` and `export(1M)` commands, or alternatively, `libmgr(1M)` or robot `robottool(1M)`). Importing and exporting media also updates the library catalog. See Chapter 3, “ASM Basic Operations”.

Note, however, that each vendor’s library handles the import and export of media 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 media into the import/export unit before exporting media from the library. Libraries like the Grau, StorageTek, and Fujitsu import/export media use their own utilities, while the `import` and `export` commands only update the catalog entries used by ASM.

Note: Make sure the system administrator understands the vendor’s system documentation and the specific StorageTek manual page for that system.

Tracking Exported Media - The ASM Historian

The ASM historian keeps track of media exported from a library or a manually mounted device. The historian acts like a virtual library; however, it has no defined hardware devices. Like a library, it is configured in the `mcf` file, has a catalog that records entries on all volumes associated with it, can import and export volumes, and appears in `robottool` as another ASM library.

The historian is optionally 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  /etc/fs/samfs/sam_historian
```

where *n+1* is the last equipment ordinal in the `mcf` file plus 1. If you want to use a different equipment ordinal or pathname for the catalog, you only need to define the historian in the `mcf`.

The historian library catalog is initialized with 1000 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 volumes that have been exported from the library. In this case, you need to build a catalog from the existing volumes as described in the `build_cat(1M)` manual page.

Two configuration parameters in `defaults.conf` affect the behavior of the historian. If `exported_media = unavailable`, then any media exported from a library is set to unavailable to the historian. Requests for media flagged as unavailable receive an ESRCH error.

The `attended` parameter tells the historian if an operator is available to handle mount requests. If `attended = no`, an operator is unavailable and an error is returned. See the `historian(7)` and `defaults.conf(4)` manual pages for more detailed configuration information.

Ampex Media Libraries

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

The Ampex libraries are configured like other SCSI interface libraries. The tape drivers and runtime library are supplied by Ampex and must be installed before installing ASM. ASM requires version 3.4 of the Ampex DST Tape Device Driver; earlier versions do not work. LSC provides a separate package for Ampex support (LSCdst). Install this package following the procedures described in Step 3: “Add the Packages”, Chapter 2, “ASM Installation and Configuration”.

Configuration

To configure an Ampex library for use with ASM, you need to properly identify the system in the `/etc/fs/samfs/mcf` file. The following example fragment configures a single D2 tape drive with a robot. The `mcf` entries appear as:

```
# Equipment          Eq   Eq  Family   Dev  Additional
# Identifier         Ord  Ty  Set      St   Parameters
#
/dev/samst/c5t6u0    55   rb  am55     on   /usr/tmp/ampx812_cat
/dev/rdst14,1        56   d2  am55     on   /dev/rdst14,7
```

The first entry is almost identical to the systems configured in Chapter 2, "ASM Installation and Configuration". In Step 9 of Chapter 2, we configured two robots in a similar manner, as follows:

1. 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`).
2. Determine an equipment ordinal (`55`).
3. Define the equipment type (`rb`).
4. Associate the library and drive using a common family set name (`am55`).
5. Set the device state to `on`.
6. 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 field are slightly different, due to the fact that Ampex 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*. As with other drives, these entries are also symbolic links to the hardware pathname of the device. "*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). Any other value for *b* is not accepted by ASM.

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

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

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 that you chose for the library (`am55`).

Formatting a D2 Tape

It is not necessary to use Ampex-supplied commands to format tapes since the ASM software automatically formats them. However, if they choose, users can format tapes by using the Ampex-supplied `dd2_format_tape` utility.

ASM formats the tapes using the `tplabel` command with the –`erase` option. During formatting, ASM writes several system zones to the tape. These zones contain the volume ID, which must be identical to the tape's ANSI-standard volume label. The zones allow ASM to identify the mounted tape without rewinding it to the beginning to read the ANSI-standard label.

ASM can relabel formatted D2 tapes by using the `-erase` option from the `tplabel` command, or by checking the `erase` option in the label popup when using `devicetool`. 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 re-label 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.

Ampex Operations

When ASM takes control of the system, the buttons on the front panel of the library are *NOT* disabled as they are for other libraries. Use the ASM utilities to perform importing and exporting.

Caution: Do not use the buttons on the front of the Ampex libraries to load and unload tapes. This confuses ASM and the library.

To export a tape, use the unload command from either `libmgr` or `samu`. `unload` positions the tape to a system area and allows the tape to be unloaded from the drive and made available to the operator.

ADIC Scalar 224 and 448 Libraries

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

ADIC Operations

Operations on the ADIC vary depending on whether a mailbox is available. The mailbox is used to import/export media from the library. Those operations that are different when using a mailbox are called out in this section.

Configuration Tips

The ACL Scalar 2/24 and 4/48 should not be configured with auto-clean or auto-load when running ASM. See the subsection below, "How to Clean Tape Drives" for more information.

Auto-load may be used during the initial loading of cartridges as long as ASM is not running. Remember to disable auto-load when ASM is running.

How to Import Tapes (No Mailbox)

To import tapes in an ADIC library without a mailbox, follow this sequence:

1. Turn the robot state to off.
2. Using `robottool(1M)`, select the robot, click the Change State button, and select “off”.

   OR

   Using samu`(1M)`, enter the following from the command line:

   :**off** ***eq***

   where ***eq*** is the equipment number for the robot as defined in `mcf`(4).
3. Unlock and open the door to the library.
4. Load media into available slots.
5. Close and lock the door to the library.
6. The robot re-initializes and scans the media in the library. ASM updates the catalog, which now contains the tapes VSN. The library state displays “on”.

How to Import Tapes (Mailbox Available)

To import tapes on an ADIC library with a mailbox, follow this sequence:

1. Enter the following:

   # **import** ***eq***

   OR

   Using `robottool(1M)`, select the library and click the Import button
2. Push the OPEN button on the front panel of the library. The door opens.
3. Load media into the mailbox slots.
4. Push the CLOSE button on the front panel of the library. Physically close the door to the mailbox.

   The robot moves each tape from the mailbox into an empty slot. The catalog is updated and should now contain the tape VSNs.

How to Export Tapes (No Mailbox)

To export tapes on an ADIC library without a mailbox, follow this sequence:

1. Turn the robot state to off.
2. Using `robottool(1M)`, select the robot, click the Change State button, and select “off”.

   OR

   Using samu`(1M)`, enter the following from the command line:

   :**off** ***eq***

   where ***`eq`*** is the equipment number for the robot as defined in `mcf`(4).
3. Unlock and open the door to the library.
4. Remove the tapes from their respective slots.
5. Close the door to the library and lock it.

   The robot re-initializes and scans the media in the library. ASM updates the catalog. The removed tapes are now unavailable in the catalog. The library state displays “on”.

How to Export Tapes (Mailbox Available)

To export tapes from an ADIC library with a mailbox, follow this sequence:

1. Enter the following:

   # **export -v** ***vsn eq*** *or*

   # **export -s** ***slot eq***

   `vsn` is the volume serial name of the tape. ***`slot`*** is the slot number in which the tape is located. ***`eq`*** is the equipment number for the robot as defined in `mcf`(4).

   OR

   Using `robottool(1M)`, select the library, and click the Export button.

   The robot moves the tape from the storage slots into the mailbox. ASM updates the catalog.
2. Push the OPEN button on the front panel of the library. The door opens.
3. Remove the tapes from the mailbox.
4. Push the CLOSE button on the front panel of the library. Physically close the door to the mailbox.

Sony B9 and B35 Libraries

The Sony B-9 and B35 series libraries contain 1 or 2 DTF drives with a capacity of up to 70 DTF tapes. These Stacker libraries optionally use barcode readers. The Sony 8400 PetaSite is also supported (see the next subsection).

SONY Operations

Operations on the Sony libraries may vary depending on whether a mailbox is available. The mailbox is used to import and export media from the library. This section describes operations that are different when using a mailbox.

Configuration Tips

The Sony B-9 and B-35 should not be configured with auto-clean or auto-load when running ASM. See the subsection "How to Clean Tape Drives" for more information.

Auto-load may be used during the initial loading of cartridges as long as ASM is not running. Remember to disable auto-load when ASM is running.

How to Import a Tape without a Mailbox

To import a tape on a Sony library without a mailbox, follow these steps:

1. Using samu(1M), enter the following from the command line:

   :**unload** ***eq***

   where ***eq*** is the equipment number for the robot as defined in `mcf`(4).

   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 library.
3. Load media into the available slots.
4. Close and lock the door to the library.

   The robot re-initializes and scans the media in the library. ASM updates the catalog, which should now contain the VSN of the tape. The library state is set to "on".

How to Export a Tape without a Mailbox

To export a tape on a Sony library without a mailbox, follow these steps:

1. Using samu(1M), enter the following from the command line:

   :**unload** ***eq***

   where ***eq*** is the equipment number for the robot as defined in `mcf`(4).

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

2. Unlock and open the door to the library.
3. Remove the tapes from their respective slots.

4. Close and lock the door to the library.

   The robot re-initializes and scans the media in the library. ASM updates the catalog and the tapes currently in slots and the removed tapes are left to the historian file. The library state is set to “on”.

SONY 8400 PetaSite Library

The Sony 8400 PetaSite Series library contains “N” DTF drives with a capacity up to “N” DTF tapes. The library has an eight slot import and export mailbox that can also be used as storage slots. The system uses a barcode reader.

SONY 8400 Operations

Operations on the Sony 8400 are different from other Sony models because the mailbox slots (slots 400 – 407) can also be used as storage slots. For this reason, the import and export commands are more straightforward for this system.

The ASM catalog keeps track of the mailbox slots. Therefore, when importing and exporting tapes, you must notify the catalog.

How to Import Tapes

To import tapes, follow these steps:

1. Open the door of the library by pushing the open/close button on the front panel of the library.
2. Load media into the mailbox slots.
3. Push the open/close button on the front panel of the library and manually close the door to the mailbox.

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

4. Use one of the following methods to enable ASM to recognize the imported media:
   - Enter the following command: `#` **`import`** ***`eq`***

     OR

   - Using `robottool`(1M), select the VSN and click the Import button on the right side of the display panel. .

How to Export Tapes

In order to export tapes from the Sony 8400, the tape must be in one of the mailbox slots. Note that when exporting tapes, you must notify the catalog.

If you are using the mailbox slots as storage and the tape you want to export is in one of the mailbox slots (slots 400-407), perform the following steps:

1. Push the open/close button on the front panel of the library. The door opens.
2. Remove the media from the mailbox slot.
3. Push the open/close button on the front panel of the library and manually close the mailbox door.
4. Use one of the following methods to enable ASM to recognize the exported media:
   - Enter the following command: `#` **`export`** ***`eq`***

     OR

   - Using `robottool`(1M), select the VSN and click the Export button on the right side of the display panel.

If you are not using the mailbox storage slots, perform the following steps:

1. Issue the move command to move the tape to a mailbox slot (slot 400-407).

   ```
   # move source_slot destination_slot eq
   ```

2. Push the open/close button on the front panel of the library. The door opens.
3. Remove the media from the mailbox slots.
4. Push the open/close button on the front panel of the library and manually close the door to the mailbox.
5. Use one of the following methods to enable ASM to recognize the exported media:
   - Enter the following command: `#` **`export`** ***`eq`***

     OR

   - Using `robottool`(1M), select the VSN and click the Export button on the right side of the display panel.

How to Move a Tape to a Different Slot

Use either `robottool` or the `move` command to move a tape from one slot to another. Note that the source slot must be occupied and the destination slot must be empty.

If you are using `robottool`(1M), follow these steps:

1. Select a tape VSN and click the Move button.
2. Enter the destination slot and click Move.

If you are using the `move(1M)` command, follow these steps:

1. Enter the following information:

   # **move** ***source_slot destination-slot eq***

   `source_slot` is a slot currently in use in the library.

   `destination_slot` is an unoccupied slot number in the robot.

   `eq` is the equipment number for the robot as defined in `mcf`(4).

How to Clean a Tape Drive

If autoclean is not enabled on this library, you should clean the drives periodically as requested from the tape unit. You must have a bar coded cleaning tape whose barcode VSN starts with the label “CLN” or “CLEAN”.

To request that a drive be cleaned, enter the following:

cleandrive ***eq***

`eq` is the equipment number of the tape drive as defined in `mcf`(4).

How to Audit a Library

To audit a library, use `libmgr` to select the robot from the robots display, select Audit, and follow the prompt.

How to Build a Catalog

The Sony 8400 is a bar-coded system. When ASM is initialized, it automatically builds a catalog. The process of importing and exporting tapes (described in this section) keeps the catalog current.

How to Label a Tape

If you are not using an automated library with a barcode reader, you must label the tape before using ASM. If your automated library uses barcodes, make sure that the following line is included in the `/etc/fs/samfs/defaults.conf` file. This line ensures that the six low order-digits become the VSN on the tape label.

```
labels = barcodes_low
```

If you want the six high-order digits to become the VSN on the tape label, include the following line in the `/etc/fs/samfs/defaults.conf file`:

```
labels = barcodes
```

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

Caution: Labeling or relabeling media will destroy all data on the media. Make sure this is your intent before labeling or relabeling media.

Select one of the following methods to label a new tape:

- Enter the following information:

  ```
  # tplabel  -new -vsn new_vsn -slot slot eq
  ```

  where ***new_vsn*** is the VSN for the new tape, ***slot*** is the slot number in which the tape is located, and ***eq*** is the equipment number for the robot as defined in `mcf`(4).

OR

- Use `robottool`(1M) to select a medium. Click the Label button and enter the information for the new VSN as prompted.

  The tape is mounted and positioned, and the new tape label is written.

To relabel a tape, follow these steps:

- Enter the following information:

  ```
  # tplabel  -old old_vsn -vsn new_vsn -slot slot eq
  ```

  where ***old_vsn*** is the volume serial name of the old tape, ***new_vsn*** is the VSN for the new tape, ***slot*** is the slot number in which the tape is located, and ***eq*** is the equipment number for the robot as defined in `mcf`(4).

- You can also use `robottool`(1M) to select a tape VSN and click the Label button. Enter the information old VSN and the new VSN as prompted.

  The tape is mounted and positioned, and the tape label is written.

How to Mount a Tape on a Drive

You can mount an ASM tape by using `robottool`(1M) or `samu`(1M).

To load a tape using the `robottool`, follow these steps:

1. Select the VSN you want to mount.

2. Click the Mount button.

To mount a tape from the `samu`(1M) command line interface, follow these steps:

1. Start `samu`:

   # **samu**

2. Enter one of the following `samu` commands. The first command mounts a tape given its VSN, the second command mounts a tape given its slot number. Note that the colon in `samu` brings up the command line prompt:

   `:mtvsn` ***`eq vsn`***

   `:mtslot` ***`eq slot`***

   `vsn` is the volume serial name of the tape.

   `slot` is the slot number in which the tape is located.

   `eq` is the equipment number for the robot as defined in `mcf`(4).

How to Remove a Tape From a Stuck Drive

If a tape becomes stuck in a drive, follow these steps:

1. Turn off the drives in the library.
2. Turn off the library.
3. Physically remove the tape from the drive. Make sure you do not damage either the tape or the drive.
4. Turn on the robot and the drive.
5. Adjust the catalog `(chmed -o)` to remove the damaged tape unless the robot does an audit when it is turned on.

StorageTek Libraries

This section describes the operation of SCSI-attached and ACSLS-attached StorageTek libraries. The following topics are presented:

- Configuring libraries
- Building library catalogs
- Managing catalogs while importing and exporting media
- Common error messages

Managing SCSI Attached StorageTek Libraries

The management of SCSI-attached StorageTek (STK) libraries is straightforward and similar to other SCSI-attached libraries. A SCSI-attached StorageTek library is controlled by ASM using the SCSI standard for media changers. It is not necessary to add optional ASM packages.

Configuring SCSI-attached StorageTek Libraries

Updating st.conf

To add support for the DLT and 9840 tape drives, it is necessary to update the `/kernel/drv/st.conf` file. Read the `/opt/LSCsamfs/examples/st.conf_changes` file into `/kernel/drv/st.conf` and make changes as specified in the file. For more information, see Step 4: "Add Tape Support to st.conf" in Chapter 2, "ASM Installation and Configuration".

Following is an example of a `/kernel/drv/st.conf` file that has been modified to add support for the 9840 tape drive in the Solaris kernel:

```
tape-config-list =
"STK 9840", "STK 9840 Fast Access", "CLASS_9840",
CLASS_9840 = 1,0x36,0,0x1d679,1,0x00,0;
```

mcf file

The `mcf` for SCSI-attached StorageTek libraries uses the family set name to associate drives with a library.

- The library and each associated drive are assigned a unique equipment ordinal. The equipment ordinal is a unique integer from 1 to 32757 (required field).
- The equipment identifier field is the `/dev/samst` entry for the library itself. For more information, see the example on configuring a robotically-loaded DLT drive in Step 9: "Configure ASM", Chapter 2, "ASM Installation and Configuration".
- The equipment type is a two-character mnemonic for the device type (required field). See the `mcf(4)` manual page for specific equipment types.
- The device state defines the state of the device when ASM is initialized (optional field).
- The additional parameters field must be the full pathname to the library catalog (required field).

The following `mcf` entries define an StorageTek library with two 9840 drives.

```
# Equipment            Eq   Eq   Family   Dev   Additional
# Identifier           Ord  Ty   Set      St    Parameters
#
/etc/fs/samfs/c0t3u0   50   sk   sk50     on    /etc/fs/samfs/sk50_cat
/dev/rmt/0cbn          51   sg   sk50     on
/dev/rmt/1cbn          52   sg   sk50     on
```

Building a Catalog

It is not necessary to build a catalog when working with a SCSI-attached StorageTek robot. The `robots` daemon checks for a catalog file for its library when the daemon is initialized. If a catalog file is not present, it builds the catalog. The following error message appears in sam-log when the `robot` daemon does not find a catalog file:

```
May 22 13:50:52 baggins archiver[573]: err Fatal OS
call error: open(/etc/fs/samfs/scsi_stk) called
from: schedtbl.c:197: No such file or directory
```

However, a catalog is still built. Although the VSN's capacity and use count remains at zero until the VSN is loaded into a drive, an `archiver –lv` indicates the tape capacity is known to the archiver. This is determined by the media type defined in the `mcf` file.

A full audit can also be done to update the library catalog with the correct capacity and use count. A full audit mounts each piece of media, reads the label, and updates the catalog. A StorageTek library should be audited in the following situations:

- After moving media in the library without using ASM commands.
- If you are uncertain about the status of the library catalog and want to update it (e.g., after a power outage).

To perform a full audit on a media in a StorageTek library, enter one of the following:

- From `samu(1M)`, enter the following command where *`eq`* is the ordinal of the library:

  :audit *eq*

From `robottool(1M)`, select the library in the robots panel , and then select the "Full Audit" button. The following message appears: "This command will take a long time. Do you wish to Continue or Cancel?" Select "Continue" to proceed with the audit.

Importing and Exporting Media

This section describes how to import and export media for SCSI-attached StorageTek libraries.

A *mailbox* is an area used for inserting and removing media from the library. Use *Import* to place a cartridge in the mailbox, and then use ASM to move the cartridge into a storage slot. Use *Export* to use ASM to move the cartridge from a storage slot to the mailbox. Some StorageTek libraries only import and export one piece of media at a time. A *mailslot* handles a single piece of media and is used to import/export to the library.

Typical ASM supported StorageTek libraries that use a mailbox include the StorageTek 9714 and 9710. The StorageTek 9730 uses a mailslot. In StorageTek literature, the mailbox and mailslot are often referred to as the CAP.

How to Import Tapes Using a Mailbox

To import tapes into a StorageTek library, follow this sequence:

1. Open the mailbox using method suggested for your type of StorageTek library.

2. Physically place the media into the mailbox or mailslot.

3. Close the mailbox.

4. To import the media, enter one of the following commands. The media is moved from the mailbox to a storage slot and the StorageTek library catalog is updated for each piece of media.

   From the command line enter:

   ```
   # import eq
   ```

   where *eq* is the ordinal of the library.

   From `robottool(1M)`, select the library in the robots panel and then select the “Import” button.

   From `samu(1M)` enter the following command, where *eq* is the ordinal of the library:

   ```
   :import eq
   ```

How to Export Tapes Using a Mailbox

To export (remove) tapes from a StorageTek library, follow this sequence:

1. Enter one of the following. The media will be moved from a storage slot to the mailbox or mailslot.

   Using the command line to export media by slot or VSN, enter one of the following where *slot* is the number of the slot where the media is stored, VSN is the volume serial name of the media, and *eq* is the ordinal of the library:

   #**export -s *slot eq*** *or*

   #**export -v *vsn eq***

   Using `robottool(1M)`, select the slot or VSN in the VSN catalog panel. Then select the "Export" button.

   Using `samu(1M)` to export media by slot, enter the following command where *eq* is the ordinal of the library and *slot* is the slot number in which the tape resides:

   :export *eq slot*

   The StorageTek catalog is updated as each piece of media is exported. The catalog entry for each piece of exported media is moved to the ASM historian.

2. Open the mailbox or mailslot using the method suggested for your type of StorageTek library.

Common Error Messages

Configuration errors are expected, especially during initial installation of a SCSI-attached library and ASM. Following are examples of typical error message and suggestions for problem resolution.

Example One

```
May 18 12:38:18 baggins genu-30[374]: Tape device 31 is
default type. Update /kernel/drv/st.conf?
```

This message, found in the sam-log, and its corresponding message seen in the device log for an associated drive:

```
1999/05/18 12:34:27*0000 Initialized. tp

1999/05/18 12:34:28*1002 Device is QUANTUM , DLT7000

1999/05/18 12:34:28*1003 Serial CX901S4929, rev 2150

1999/05/18 12:34:28*1005 Known as Linear Tape(lt)

1999/05/18 12:34:32 0000 Attached to process 374

1999/05/18 12:38:18 1006 Slot 1

1999/05/18 12:38:18 3117 Error: Device is type default.
Update /kernel/drv/st.conf
```

Indicates that the appropriate changes have not been made to the /kernel/drv/st.conf.

Following is a /kernel/drv/st.conf entry for a DLT7000 and an entry for a 9840 tape drive:

```
tape-config-list =
"STK 9840", "STK 9840 Fast Access", "CLASS_9840",
QUANTUM DLT7000", "DLT 7000 tape drive", "dlt7-tape",
CLASS_9840 = 1,0x36,0,0x1d679,1,0x00,0;
dlt7-tape = 1,0x36,0,0xd679,4,0x82,0x83,0x84,0x85,3;
```

Example Two

This message appears in the device log for a drive:

```
1999/05/18 11:47:53*0000 Initialized. tp

1999/05/18 11:48:09*1044 System error:
open(/dev/rmt/../../devices/pci@4,4000/scsi@4/samst@0,0:a
,raw): No such device or address

1999/05/18 11:48:09*1009 Changing state to OFF
```

In this example, the mcf is incorrect. The correct symbolic links, which point to the hardware paths, were not used. Locate the /var/adm/messages entries for this device. Next find the /dev/samst symbolic link that points to this hardware path. See the example on configuring a robotically-loaded DLT drive in Step 9: “Configure ASM” in Chapter 2, “ASM Installation and Configuration”.

Managing ACSLS-Attached StorageTek Libraries

Configuring and managing an ACSLS-attached StorageTek library is considerably more complicated than configuring and managing a SCSI-attached StorageTek library. The ACSLS software package supplied by StorageTek controls the library. ASM interfaces with the StorageTek software using the stk daemon. This daemon controls the StorageTek media changers through the ACSAPI interface. The ACSAPI package is supplied by LSC, Inc.

It is necessary to install the LSCstk package. This package supplies the stk daemon software that controls the StorageTek media changers through the ACSAPI interface. The package is supplied with the ftp package download instructions. To satisfy product dependencies, you must install sampkg first. See Step 3: “Add the Packages” in Chapter 2, “ASM Installation and Configuration”.

Configuring ACSLS-Attached StorageTek Libraries

The ASM configuration should not be attempted until the StorageTek ACSLS software package is installed and working.

Updating st.conf

It is necessary to update the `/kernel/drv/st.conf` file to add support for the DLT and 9840 tape drives. To do this, read the `/opt/LSCsamfs/examples/st.conf_changes` file into `/kernel/drv/st.conf` and make any changes as specified in the file. Also, see Step 4: “Add Tape Suport to `st.conf`” in Chapter 2, “ASM Installation and Configuration”.

Following is an example of a `/kernel/drv/st.conf` file that has been modified to add support in the Solaris kernel for both the 9840 tape drive and the DLT7000 tape drive:

```
tape-config-list =

  "STK 9840", "STK 9840 Fast Access", "CLASS_9840",

  "QUANTUM DLT7000", "DLT 7000 tape drive", "dlt7-tape",

CLASS_9840 = 1,0x36,0,0x1d679,1,0x00,0;

dlt7-tape = 1,0x36,0,0xd679,4,0x82,0x83,0x84,0x85,3;
```

mcf file

The `mcf` for ACSLS-attached StorageTek libraries uses the family set name to associate drives with a library.

- The library and associated drives are assigned a unique equipment ordinal. The equipment ordinal is a unique integer from 1 to 32757 (required field).
- The equipment type is a two-character mnemonic for the device type (required field). See the `mcf(4)` manual page for specific equipment types.
- The device state defines the state for the device at the time ASM is initialized (optional field).
- The additional parameters field must be the full pathname to the library catalog (required).
- An additional parameters file is needed for ACSLS-attached libraries. This parameters file defines to ASM the system characteristics of the StorageTek library and its drives. The equipment identifier field is the full path name to an STK parameters file.

The following `mcf` entries define a StorageTek library with two associated 9840 drives:

```
# Equipment              Eq    Eq   Family     Dev   Additional
# Identifier             Ord   Ty   Set        St    Parameters
#
/etc/fs/samfs/stk50      50    sk   sk50       on    /etc/fs/samfs/sk50_c
                                                     at
/dev/rmt/0cbn            51    sg   sk50       on
/dev/rmt/1cbn            52    sg   sk50       on
```

StorageTek parameters file

Each line of the StorageTek parameters file must begin with a keyword or a comment. The keywords are:

`access`

user_id used by StorageTek for access control. If the access parameter is not supplied, the access control string is a null string. This indicates no *user_id*.

`hostname`

Hostname running the STK CSI.

`portnum`

Used for ASM and ACSLS communications

`capacity`

Sets the capacity of the media supported by StorageTek. The parameters are a list of *index* = *value* pairs separated by commas and enclosed in parentheses. *Index* is the index of the media_type file supplied by StorageTek and located in the ACSLS directory `/export/home/ACSSS/data/internal/mixed_media/media_types.dat`.

value is the capacity of the media type in units of 1024 bytes. ASM has defaults for indexes 0-12 that were current at the time of release 3.3.1. In general, it is only necessary to supply a capacity entry for an index of new media types, or to over ride the StorageTek-supported capacity.

The ASM 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

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 library follows the `device_path_name`. This description starts with an open parenthesis followed by four `keyword = value` pairs and a closed parenthesis. The `keyword = value` pairs may be separated by a comma, colon or a space. Use the information supplied by the ACSLS `query drive` command to configure the `device_path_name`.

Acs	ACS number for drive as configured in the StorageTek library
lsm	LSM number for drive as configured in the StorageTek library
panel	PANEL number for drive as configured in the StorageTek library
drive	DRIVE number for drive as configured in the StorageTek library

Following is an example of a parameters file for a StorageTek library.

```
#
#  This is file: /etc/fs/samfs/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)
```

ssi.sh script

ACSLS-attached StorageTek libraries require that you edit the `ssi.sh` script supplied with the StorageTek package. The library 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 the 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 script can be found in `/opt/LSCsamfs/examples/ssi.sh.` You must copy this example script or create your own ssi.sh script in `/etc/fs/samfs/ssi.sh.` The `ssi.sh` script is not automatically created during installation of the `samstk` package.

A sample `ssi.sh` file follows:

```
#! /bin/csh
#
set old_ssi = ‘/bin/ps -e | grep ssi_so| cut -c1-6‘
if (“$old_ssi” != “” )then
 kill –TERM $old_ssi
endif
setenv CSI_TCP_RPCSERVICE TRUE
setenv CSI_UDP_RPCSERVICE TRUE
setenv CSI_CONNECT_AGETIME 172800
setenv CSI_RETRY_TIMEOUT 4
setenv CSI_RETRY_TRIES 5
setenv ACSAPI_PACKET_VERSION 4
/etc/fs/samfs/ssi_so $3 50014 23
```

Building a Catalog

The ASM administrator must create the initial catalog for ACSLS-attached libraries. Unlike a SCSI-attached StorageTek library, an ASM catalog is not created automatically. Assuming a system with a StorageTek library and an `mcf` file entry as such:

```
# Equipment            Eq   Eq   Family    Dev   Additional
# Identifier           Ord  Ty   Set       St    Parameters
#
/etc/fs/samfs/stk50    50   Sk   sk50      on    /etc/fs/samfs/sk50_cat
```

An empty catalog with 1000 entries is built using the command:

```
server# build_cat –s 1000 /etc/fs/samfs/sk50_cat < /dev/null
```

Then, bring up `sam-init` and issue a series of `import` commands to fill the catalog. Note that the media must be physically present in the ACSLS-attached StorageTek library for the import commands to be successful.

```
server# import –v volser1 sg 50
server# import –v volser2 sg 50
server# import –v volser3 sg 50
```

An alternate method can be used for large libraries with many volumes. The `build_cat` command can be given a file containing a list of VSNs that it uses to populate the initial catalog. See the `build_cat` manual page for more information. Note that the slot position of the tape in the ACSLS-attached StorageTek library has no relationship to the slot number of the VSN in the ASM catalog. Following is an example file with a list of VSNs to populate a library. The first column is the ASM catalog slot number, followed by the label, the bar code and then the media type.

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

The `audit` command is not supported for ACSLS-attached robots.

Importing and Exporting Media

This section describes how to import and export media for ACSLS-attached StorageTek libraries.

A *mailbox* is an area used for putting media inserting into and removing media from the library. Use *import* to place a cartridge in the mailbox, and then use ASM to move the cartridge into a storage slot. Use *export* to use ASM to move the cartridge from a storage slot to the mailbox. Some StorageTek libraries only import and export one piece of media at a time. A *mailslot* handles a single piece of media and is used to import and export to the library.

Typical ASM supported StorageTek libraries that use a mailbox include the StorageTek 9714 and 9710. The StorageTek 9730 uses a mailslot. In StorageTek literature, the mailbox and mailslot are often referred to as the CAP.

Note: When importing and exporting media from an ACSLS-attached robot, ASM commands only influence the ASM catalog. The media is not physically inserted into or removed from the library. You must use the ACSLS commands to physically move the media. It is the system administrator's responsibility to keep the ACSLS inventory and the ASM catalog in sync.

How to Import Tapes

1. From the command line enter:

   `# `**`import –vsn volser1`** ***`eq`***

   where *`eq`* is the ordinal of the library.

   – The new VSN appears in the ASM catalog. If the VSN was in the historian, ASM will move the VSN information from the historian to the catalog.

How to Export Tapes Using a Mailbox

From the command line, media can be exported by slot or VSN. Enter one of the following where *`slot`* is the number of the slot where the media is stored, *`vsn`* is the volume serial name of the media, and *`eq`* is the ordinal of the library:

```
#export -s slot eq     or
#export -v vsn eq
```

From `robottool(1M)`, select the slot or VSN in the VSN catalog panel. Select the "Export" button.

From `samu(1M)`, you export media by slot number. Enter one of the following commands where *`eq`* is the ordinal of the library and *`slot`* is the slot number where the tape resides:

```
:export eq slot
```

ASM updates the StorageTek catalog as each piece of media is exported, and moves the catalog entry for each piece of exported media from the ASM library catalog to the ASM historian.

Example One

```
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
```

In this example, the `samstk` package has not been installed. See Step 3: “Add the Packages” in Chapter 2, “ASM Installation and Configuration Procedure”.

Example Two

```
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.
```

Check the stk parameters file for syntax errors. Remember that each line must begin with a keyword or a comment. See the `stk(7)` manual page for more details on the stk parameters file.

Example Three

```
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
```

And finally:

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

The `samu r` display displays:

```
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. Can you ping the hostname?

Check `portnum` in the `stk` parameters file. In ACSLS 5.3, the default port number, 50004 is now used for a different application. Try a higher port number (e.g., 50014).

Example Four

```
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

Here the `import` command was used to import a VSN to the ASM catalog but the VSN is not in the StorageTek library. The medium must be present in the ACSLS-managed library before the ASM command can be successful.

Chapter 5. ASM File Systems

Introduction

The ASM file system presents a standard UNIX file system interface to users. Although the ASM command set enhances standard UNIX commands, it runs with no changes to the UNIX kernel or users' programs. The SAM-QFS file system can be selected for enhanced features and performance.

This chapter presents the following topics:

- File system features
- Volume management
- File attributes
- File system commands and utilities
- File system operations

File System Features

Vnode Interface

ASM is implemented using the standard Sun Solaris virtual file system (`vfs/vnode`) interface. The kernel first intercepts all requests for ASM resident files. The file is identified as ASM resident and the request is passed to the ASM file system. By using the `vfs` interface, ASM works with the standard Solaris kernel and requires no modifications within the kernel for the file management support. Thus, ASM is protected from operating system changes and does not normally require extensive regression testing when the operating system is updated or improved.

File System Size

ASM handles all requests for ASM-resident files. The ASM file system is type `samfs`. Multiple file systems are supported. Each file system supports up to 200 disk partitions for virtually unlimited storage capacity.

Volume Management

The method of configuring file systems and disk partitions, including entire disks, in the master configuration file (mcf) controls the ASM volume management ASM knows the topology of the devices it controls. This is in contrast to most UNIX file systems that can only address 1 device. This means ASM does not require a special add-on volume manager.

File System Geometry

Disk space is allocated in disk allocation units (DAUs), the basic unit of online disk storage. While sectors and blocks describe the physical disk geometry, the DAU describes the file system geometry.

For improved performance and magnetic disk usage, ASM uses a variable disk allocation unit (DAU) with two sizes: small DAUs and large DAUs. The small DAU is 4 kilobytes and the large DAU is 16, 32, or 64 kilobytes. The available DAU size pairs are: 4/16, 4/32, and 4/64. Depending on the type of file data stored on the disks, selecting the larger DAU pairs may improve the file system performance significantly.

ASM first maps the file to small DAUs. When the file exceeds 8 small DAUs, ASM maps the remaining file blocks to large DAUs. The result is better performance along with more complete usage of the magnetic disk.

Each file system has a DAU setting. Thus, several mounted file systems may be active on a server, each with a different DAU setting. The DAU setting is determined when the file system is created using `sammkfs(1M)`; it cannot be changed dynamically. The default DAU setting is 4/16k.

In contract to ASM's large DAU, ASM-QFS supports a fully adjustable DAU. SAM-QFS DAU is adjustable from 16 to 65535 1024-byte blocks. An adjustable DAU is useful for tuning the file system with the physical disk storage device, eliminating the system overhead caused by read, modify, and write operations. Applications that benefit from this feature use large block sequential I/O.

File System Allocation

ASM supports striping or round-robin disk allocation. Striping spreads the file allocation across the file system devices while round robin allocates files on individual file system devices. Striping is used when performance for 1 file requires the additive performance of all the devices. Round robin is useful for multiple data streams because the aggregate performance can actually exceed striping for this type of environment.

File System Inodes

There is no pre-defined limit for the number of files on an ASM file system. Since the inodes (information about the files) are dynamically allocated, the maximum number of files is limited only by the amount of disk storage comprising the file system. The `inodes` are cataloged in the `.inodes` file under the mount point. The `.inodes` file requires 512 bytes of storage per file. SAM-QFS inodes are located on the meta device(s) and are separated from the file data devices.

Large File Size

ASM is designed to accept very large files up to $2^{64}-1$ bytes ($2^{34}-1$ Gbytes). ASM is able to process large file sizes because it is fully 64-bit compatible.

Note: ASM data is written to the archives using standard tar format. For disaster recovery purposes, we provide the `gnutar` command for restoring data on UNIX systems. Note, however, that `gnutar` does not presently work with files larger than 8 Gbytes.

Support for Special Files

Support for special devices is available for command interfaces that require special devices. It is suggested that you create a symbolic link to the physical name of the device in the `/devices` subtree rather than using `mknod`. See the `ftp(1)` in the `.ftpd(1M)`, `mknod(2)`, and `symlink(2)` manual pages for more information.

Fast File System Recovery

One of the key functions of a file system is to recover quickly after an unscheduled boot. Standard UNIX file systems require a lengthy file system check (fsck) to repair inconsistencies after a system failure. Running a fsck on a terabyte-size file system can take hours.

ASM does not require file system checks and recovers from system failures without using journaling. It accomplishes this by dynamically utilizing identification records, serial writes, and error checking for all critical I/O. Just seconds after a system failure, ASM is online again for a terabyte-size file system.

Volume Management

ASM volume management uses the Solaris physical device drivers to pass I/O requests to and from the underlying devices. The ASM volume manager groups physical devices into family sets on which the ASM file system is mounted.

ASM Volume Management

For ASM, family sets are defined using two device types: ms and md. These device types are defined using the equipment type field in the master configuration file, `/etc/fs/samfs/mcf`. A sample `mcf` file for ASM is shown below:

```
# ASM file system configuration example
#
# Equipment       Eq    Eq   Fam.  Dev.    Additional
# Identifier      Ord  Type  Set   State   Parameters
#-----------      ---   --  ------ ------  ---------------
samfs1             10  ms  samfs1
/dev/dsk/c1t1d0s2  11  md  samfs1  -   /dev/rdsk/c1t1d0s2
/dev/dsk/c1t2d0s2  12  md  samfs1  -   /dev/rdsk/c1t2d0s2
/dev/dsk/c1t3d0s2  13  md  samfs1  -   /dev/rdsk/c1t3d0s2
/dev/dsk/c1t4d0s2  14  md  samfs1  -   /dev/rdsk/c1t4d0s2
/dev/dsk/c1t5d0s2  15  md  samfs1  -   /dev/rdsk/c1t5d0s2
```

Device	Equipment Identifier	Description
Family Set Device	**ms**	This device is used to specify the ASM file system type.
Striped or Round-Robin Devices	**md**	Data is striped or round robined across these devices. The stripe width is a mount option. The default stripe width is 1, indicating that ASM uses a stripe width of 1 DAU.

Metadata (e.g., inodes, directories, allocation maps) and file data on ASM file systems are located on the same disk. Data files are striped or round-robined through each disk partition defined within the same file system.

SAM-QFS Volume Management

For SAM-QFS, family sets are defined using the following device types: ma, mm, mr, and g*XXX*. These device types are defined using the equipment type field in the master configuration file `/etc/fs/samfs/mcf`. A sample `mcf` file is shown below with two SAM-QFS file systems:

```
# SAM-QFS file system configuration example
#
# Equipment       Eq    Eq   Fam.  Dev.    Additional
# Identifier      Ord  Type  Set   State   Parameters
#-----------      ---   --  ------ ------  ------------------
samfs1             10  ma  samfs1
/dev/dsk/c0t1d0s2  11  mm  samfs1  -   /dev/rdsk/c0t1d0s2
/dev/dsk/c1t1d0s2  12  mr  samfs1  -   /dev/rdsk/c1t1d0s2
/dev/dsk/c1t2d0s2  13  mr  samfs1  -   /dev/rdsk/c1t2d0s2
/dev/dsk/c1t3d0s2  14  mr  samfs1  -   /dev/rdsk/c1t3d0s2
/dev/dsk/c1t4d0s2  15  mr  samfs1  -   /dev/rdsk/c1t4d0s2
/dev/dsk/c1t5d0s2  16  mr  samfs1  -   /dev/rdsk/c1t5d0s2
samfs2             20  ma  samfs2
/dev/dsk/c3t1d0s2  21  mm  samfs2  -   /dev/rdsk/c3t1d0s2
/dev/dsk/c2t1d0s2  22  g0  samfs2  -   /dev/rdsk/c2t1d0s2
/dev/dsk/c2t2d0s2  23  g0  samfs2  -   /dev/rdsk/c2t2d0s2
/dev/dsk/c2t3d0s2  24  g1  samfs2  -   /dev/rdsk/c2t3d0s2
/dev/dsk/c2t4d0s2  25  g1  samfs2  -   /dev/rdsk/c2t4d0s2
```

Device	Equipment Identifier	Description
Family Set Device	**ma**	This device is used to specify the SAM-QFS file system type.
Metadata Device	**mm**	This device is used for metadata only. File data is not written to the device. You can specify multiple metadata device specifications.
Striped or Round-Robin Devices	**mr**	Data is striped or round robined across these devices. The stripe width is a mount option. The default stripe width is 0, indicating that SAM-QFS uses round robin as default.
Striped Groups	g*XXX*	A *striped group* is a logical group of devices that are striped as a unit. Data is striped or round robined across these striped groups. Groups are named g0-g127. All members in a *striped group* must be the same size.

Metadata (e.g., inodes, directories, allocation maps) on SAM-QFS file systems is located on the metadata device(s) and is separated from the file data devices. Data files are striped or round-robined through each data disk partition (mr or gXXX) defined within the same file system.

Round-Robin Allocation

ASM uses a round-robin allocation method to write one data file at a time to each successive device in the family set. When the number of files written equals the number of devices defined in the family set, ASM starts over again with the first device. Figure 5-1 depicts a file system using five round-robin devices. In this figure, file 1 is written to disk 1, file 2 is written to disk 2, file 3 is written to disk 3, and so on. When file 6 is created, it is written to disk 1, starting the round-robin allocation scheme over again.

For large files that exceed the size of the physical device, the first portion of the file is written to the first device and the remainder of the file is written to the next device.

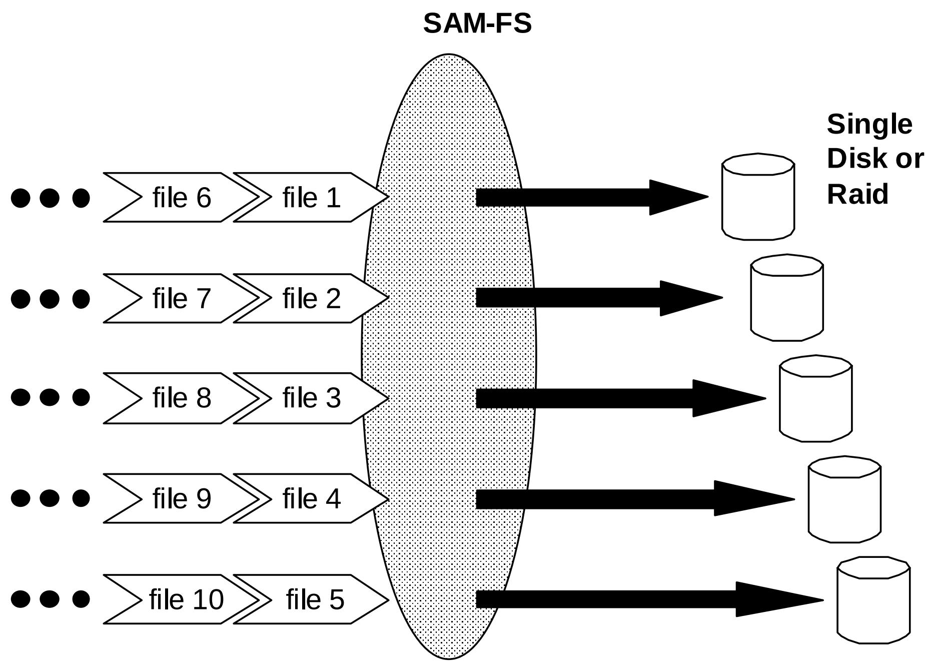


Figure 5-1 An ASM file system using round-robin allocation

Stripe Allocation

ASM uses a stripe allocation method to spread file data over all the devices in the family set. The stripe width is the DAU, which is set at file system initialization time. It is possible to increase this width at mount time with the stripe parameter. Figure 5-2 depicts a file system using five striped devices. In this figure, DAU bytes of file 1 is written to disk 1, DAU bytes of the next file is written to disk 2, DAU bytes of the next file is written to disk 3, and so on. Striping spreads all files across all the disks in DAU widths. The order of the stripe is FIFO for the files. Striping spreads the I/O load over all the disks.

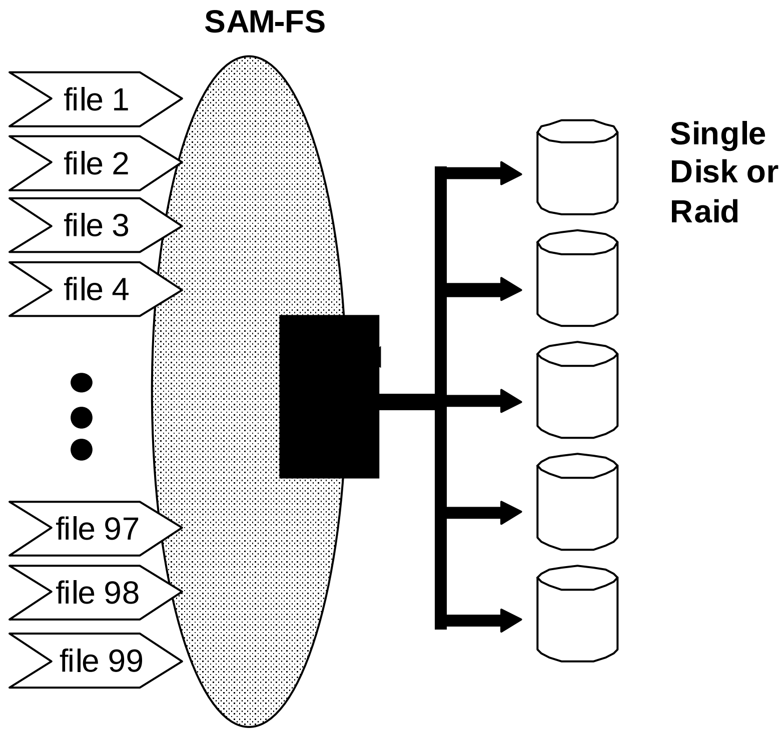


Figure 5-2 An ASM file system using standard striping

File Inode

An ASM inode is a 512-byte block of information that describes the attributes and characteristics of a file or directory. This information is allocated dynamically within the file system. ASM inodes are stored in the `.inodes` file located under the file system mount point. ASM .inodes file resides on the same physical devices(s) with the file data. In contrast, SAM-QFS .inodes file resides on the metadata devices(s) and is separated from the file data.

Like the Solaris inode, the ASM inode contains the file's POSIX inode times: access, modification and inode changed times. ASM adds a creation time, attribute change time, and a residence time. The following times are located in the inode:

`access`	Time the file was last accessed.
`modification`	Time the file was last modified.
`changed`	Time the standard UNIX inode information was last changed.
`attributes`	Time the ASM attributes were last changed.
`creation`	Time the file was created.
`residence`	Time the file changed from offline to online or vice versa.

File Attributes

ASM attributes are stored in the inode. The user sets these attributes with the user commands archive, stage, release, csum, or setfa. These attributes can also be set by the program with corresponding API functions. The following ASM attributes are listed:

`archdone`	Indicates the file's archive requirements have been met. There is no more work archiver can do on the file. Note: `archdone` does not necessarily indicate the file has been archived.
`archive –n`	The file is marked never archive (super-user only).
`damaged`	The file is damaged.
`directio`	The file is marked for direct I/O. (See the Solaris 2.6 or Solaris 7 `directio(3C)` documentation for more information.)
`offline`	The file is offline.
`release –a`	This file is marked for release as soon as one copy is made.
`release –n`	This file is marked never release.
`release –p`	The file is marked for partial release. `partial=nk` indicates the first n kilobytes of disk space will be retained for the file.
`stage –n`	The file is marked never stage.
`stage –a`	The file is marked for associative staging.

ASM attributes can be set on directories. Files that are created in a directory inherit all the directory attributes at the time of creation. Files created before an attribute is applied to the parent directory do not inherit that directory attribute.

File Display

The following example is the detailed output for the `sls` command. It displays the inode information for the file mickey.gif:

```
mickey.gif:
mode: -rw-r--r--  links:   1  owner: root      group: other
length:    319279  inode:     1407
offline;  archdone;  stage -n;
copy 1: ---- May 21 10:29     1e4b1.1    lt DLT001
access:      May 21 09:25  modification: May 21 09:25
changed:     May 21 09:26  attributes:   May 21 10:44
creation:    May 21 09:25  residence:    May 21 10:44
```

The first line indicates the file's mode or permissions, the number of hard links to the file, the owner (or user) of the file, and the group of the file.

The second line indicates the file's length in bytes and the inode number.

The next line displays the ASM file attributes and is formatted with descriptive text.

The archive copy line is displayed only if there is an active or stale copy. Following is an example of the archive copy line output:

```
copy 1: ---- Sep 11 10:43    3498f.1    mo OPT001
```

The first field indicates the archive copy number.

dash 0	stale or active entry `S` The archive copy is stale indicating the file was modified and this archive copy is a previous version of the file. `-`The archive copy is active and valid.
dash 1	archive status `r` The archive copy is scheduled to be rearchived by the archiver. `-`The archive copy is not to be rearchived by the archiver.
dash 2	unused
dash 3	damaged or undamaged status `D` The archive copy is damaged. The archive copy is not staged. `-`The archive copy is not damaged. It is a candidate for staging.

The next field shows the date and time the archive copy was written to the media.

The first hex number, 3498f, indicates the position of the beginning of the archive file on the media. The second hex number is the file byte offset divided by 512 of this copy on the archive file. For example, 1 means this is the first file on the archive file because it is offset by 512 bytes, which is the length of the tar header.

The last two fields indicate the media type and the volume serial name where the archive copy resides.

The `checksum` line is displayed only if the file has a checksum-related attribute set (i.e., generate, use, or valid). The format of the checksum line is:

```
checksum:  gen  use  val  algo:  1
```

The above line is displayed if all three checksum attributes are set for a file. If the generate attribute is not set, `no_gen` appears in place of `gen`. Similarly, if the `use` attribute is not set, `no_use` appears. `val` is displayed if a valid checksum value exists for the file; if not, `not_val` appears. The keyword `algo` precedes the numeric algorithm indicator that specifies which algorithm is used to generate the checksum value.

File System Commands and Utilities

Following is a listing of the file system commands available with ASM:

sammkfs(1M)

Constructs a new ASM file system. Use the –`r` option to restore an inodes file.

mount_samfs(1M)

Mounts an ASM file system.

samgrowfs(1M)

Expands an ASM file system by allowing the addition of logical disks.

samfsck(1M)

Checks and optionally repairs an ASM file system.

samfsdump/samfsrestore(1M)

Dumps and restores an ASM file system.

samncheck(1M)

Generates pathnames versus inode numbers for ASM file systems.

sfind(1)

Locates files in an ASM file system. `sfind` is patterned after `gnu find`.

sls(1)

Lists files in an ASM file system. `sls` is patterned after `gnu ls`.

setfa(1)

Sets file attributes for files in an ASM file system.

sam_advise(3)

Sets file attributes from the program.

For a complete description of the ASM commands and options, see Appendix C, “ASM Manual Pages”.

File System Operations

How to Mount an ASM File System

ASM mount options are used to manipulate file system characteristics. These options can be specified as parameters in the `samfs.cmd` file, on the `mount_samfs(1M)` command, or as options in the additional parameters field in the `/etc/vfstab`. Mount options can be temporarily set by using the ASM operator utility `samu(1M)`.

It is recommended that you specify mount options as parameters in `samfs.cmd`, the ASM file system command file. These parameters take precedence over parameters specified on the `mount` command or in the `/etc/vfstab` file. See the manual pages for more information.

How to Unmount an ASM File System

If you experience difficulties unmounting because the file system is busy enter:

```
server# user –c /mountpoint
```

The processes IDs that have files that are busy under the specified mountpoint are returned. these processes can be killed with the following commnad:

```
server# user –c –k /mountpoint
```

At this point you can unmount the system with the `umount` command.

How to Check File System Integrity

ASM writes validation records in all file system critical records–directories, indirect blocks, and inodes. If a directory corruption is detected, the errno EDOM is returned and the directory is not processed. If an indirect block is not valid, the errno ENOCSI is returned and the file is not processed.

EDOM	Argument is out of domain
ENOCSI	No CSi structure is available

Inodes are also validated and cross checked with directories.

The system administrator should watch the `/var/adm/sam-log` file for the above errors. The system administrator should watch the `/var/adm/messages` file for device errors. If a discrepancy is noted, the file system should be unmounted and checked with the command `samfsck`.

```
server# samfsck –V family_set_name
```

Nonfatal errors returned by `samfsck` are preceded by NOTICE. Nonfatal errors are lost blocks and orphans. The file system is still consistent if NOTICE errors are returned. These nonfatal errors can be repaired when it is convenient.

Fatal errors are preceded by ALERT. These errors include duplicate blocks, invalid directories, and invalid indirect blocks. The file system is NOT consistent if these errors occur. It is recommended that the system administrator notify LSC if the ALERT errors cannot be explained by a hardware malfunction.

How to Repair a File System

If the `samfsck` command detects file system corruption by returning ALERT messages, the reason for the corruption should be determined. If hardware is faulty, it should be repaired first. Then the file system should be repaired by specifying the –`F` option.

```
server# samfsck-F –V family_set_name
```

Chapter 6. Archiver Operations

Introduction

Archiving is the process of copying files from ASM file systems to removable media. This chapter describes the theory and operations of the archiver, provides general guidelines for developing archive policies for your site, and demonstrates how to implement policies by creating an `archiver.cmd` file (the commands file that controls the actions of the archiver).

The following topics are presented:

- Theory of operations
- The archiver commands file
- `archiver.cmd` examples
- Archive policies and guidelines
- Troubleshooting the archiver

The Archiver - Theory of Operations

The archiver periodically scans ASM file systems, examining the status of the files. As files are created and modified, the archiver copies them to removable media. Archived files are recorded in `tar` format to ensure data compatibility and mobility with other Solaris and non-Solaris systems.

The archiving process also copies the data necessary for ASM file system operation. This data consists of directories, symbolic links 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.

Typically, files are archived to removable media residing in a robot. Operator intervention is not required to archive and stage the files. Each cartridge is identified by a unique code or name, referred to as the Volume Sequence Number (VSN). To identify an individual medium, the media type and the VSN must be specified.

The archiver copies files to removable media in the standard `tar` format. This format includes the file access data (inode) and the path to the file. If a complete loss of the ASM system occurs, the tar format allows file recovery using standard operating system commands.

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 system administrator can set the archive age for files. This is necessary because normal UNIX operations (especially when using NFS) do not indicate when a file is completed. A user can indicate completion of file data by working on files in an ASM directory that is not archived, and then moving the files to a different ASM directory to be archived. The system administrator sets the no_archive archive set in the archiver.cmd file to flag files that should not be archived.

Archive Sets

The archiver organizes the files of all file systems into archive sets. An *archive set* is a group of files that is associated with a collection of removable media. Archive sets may be defined across any group of file systems. An archive set can also contain files that have different file characteristics. Archive set names are determined by the administrator and are virtually unlimited with the following exceptions: there is a reserved archive set named `no_archive`; archive sets named after each ASM file system are reserved for control structure information; and archive set names are limited to 29 characters.

The archive set named `no_archive` is defined by default. Files selected to be in this archive set are never archived.

ASM provides a default archive set for each file system. For each file system, both the file system data and data files are archived. 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 the ASM file system defined in Chapter 2, “ASM Installation and Configuration”.

Each archive set may have one to four archive copies defined. An archive set and a copy number become a synonym for a collection of media and VSNs.

In any given archive interval, all the files belonging to an archive set that need to be archived are copied to a VSN together in `tar` format to an *`archive file`*. To maximize performance, each archive file is filled with files until a certain size is reached. This size is called archmax. Its value is media-dependent and may be set by the system administrator. If files exceed this size, they are archived alone in an archive file. For optical media, the file identifier is set to the name of the archive set.

If archive requests need to be split among drives, the `–drivemin min_size` command parameter is used with the `–drives` command to set the minimum size for splitting the archive requests. See the sub-section on “Splitting Archive Requests among Drives” in this chapter for more information.

Archive Log Files and Event Logging

The 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. The system administrator sets the name of this file. It is not produced by default. For more information on the log file, see the `archiver.cmd(4)` manual page. Following is a sample line from an archiver log file with definitions of each field.

```
A 96/05/01 15:41:31 mo v1 set_1.1 d2e.b samfs1
176942 161.5 dir1/dir3/file0
```

Field	Description	Example

1	Indicates activity	`A = archive`
		`U = unarchive`
		`1-16`

2	Date (yy/mm/dd)	`99/01/01`
3	Time (hh:mm:ss)	`15:41:31`
4	Media type	`mo`
5	VSN	`v1`
6	Archive set name.copy	`set_1.1`
7	Tarfile position on medium (hexadecimal)	`d2e`
8	File offset in tarfile (in units of 512 bytes)	`b`
9	File system name	`samfs1`
10	Inode number	`161`
11	Inode generation	`5`
12	File length in bytes. If volume overflow is used, this indicates section length in bytes.	`176942`
13	Pathname	`dir1/dir3/file0`

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

Archiver Processes

The archiver consists of three programs:

- `archiver`
- `arfind`
- `arcopy`

The archiver process is responsible for scheduling the archiving activity. The `arfind` process assigns files to be archived to archive sets. The `arcopy` process copies the files to be archived to the selected media.

`archiver` is started by `sam-init` when ASM activity begins. It reads the archiver command file and builds the tables necessary to control archiving. It starts an `arfind` process for each mounted file system; likewise, if a file system is unmounted, the associated `arfind` process is stopped. `archiver` then monitors `arfind` and processes signals from an operator or other processes.

`arfind` examines the assigned file system by initially traversing the directory tree and by subsequently scanning the `.inodes` file for each file system. An archive set is identified for each file in the file system. The archive status and the file's age are used to determine if the file needs archiving. Files to be archived are placed in a list associated with an archive set copy.

`arfind` sorts the list of files by size, largest first. Optionally, arfind also sorts by age with the oldest first. Two passes are performed over the list of files. The first pass processes on-line files and the second pass processes off-line files. A batch of files is picked that fits in `archmax`, the maximum size for the media, unless associative archiving is selected (see the "Archive Set Parameters" section). If the files are off-line, they all are staged. (This must be done in case the off-line files are on the reverse side of the archive media.) The media is opened and the files are copied to it.

When the file system has been examined, `arfind` finds a VSN for each archive set copy and determines if a drive is available for that VSN. If so, an `arcopy` is started for that archive set copy using the assigned VSN. `arfind` then transfers the list of files to be archived to `arcopy.`

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 media. Status changes include modification since selection by `arfind,` file open for write, and file removed.

The archive information is entered into the inode for each file successfully copied. 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. `arcopy` then exits.

As each `arcopy` exits, `arfind` notes that the archive operation has finished and releases drives and VSNs that are reserved. When all of its archive operations have finished, `arfind` sleeps for the archive interval. The *archive interval* then is the dead time between `arfind` operations.

Archive copies to be made from offline files are ordered by VSN. This ensures that all files (within each Archive Set) on the same VSN are staged at the same time in the order in which they were stored on the media. Sites that make archive copies of many offline files should realize significant performance improvements.

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.

The Archiver Commands File

By default, the archiver runs whenever `sam-init` is started and the ASM file systems are mounted. The default settings for the archiver are:

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

Generally, a site prefers to customize the actions of the archiver to meet the archiving requirements of their site. These actions are controlled by commands located in the archiver command file: `/etc/fs/samfs/archiver.cmd.`

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

archiver.cmd Syntax

The archiver command file consists of several types of commands:

- General commands
- Archive set assignments
- Archive set parameters
- VSN pool definitions
- VSN associations

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

The following example shows a sample `archiver.cmd` file. The comments on the right hand side indicate the different type of commands as listed above. The following sections describe the commands in general; see the `archiver(1M)` manual page for details about the commands.

```
interval = 30m                                        # General Commands
logfile= /var/adm/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 Parameters
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
```

```
interval = 30m                                  # General Commands

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 optic 08b

system.2 lt ^TAPE4 [0-1]

samfs2_all.1 mo .*

samfs2_all.2 lt .*

endvsns
```

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

If errors are found in the archiver command file, the archiver logs the count of errors and displays the message `"*** No archiving will be performed.  ***"`. Then archiver waits to be restarted, which occurs when the command file changes or the archiver receives a `SIGINT`.

When the archiver receives a `SIGINT` or the command file changes, all activities including in-process archive copies, are terminated. The archiver is restarted by `sam-init`.

Whenever you make changes to the `archiver.cmd` file, you should check for syntax errors using `archiver(1M)`. Specifying `archiver(1M) -A` (all options listed) evaluates an archiver command file against the current ASM system and produces a listing of the archiver commands file, VSNs, file system content, and errors to standard out. Errors prevent the archiver from running. `archiver(1M)` can be run on an archiver file-in-progress before moving the file to /etc/fs/samfs/archiver.cmd. You can also run `archiver(1M)` without an input file in which case archiver information is generated from `archiver.cmd.` If there is not an `archiver.cmd`, the system defaults are returned. See `archiver(1M)` for a complete description of this command.

General Commands - Controlling Overall archiver Operation

General commands control the overall archiver operation. General commands are identified either by the equal sign (=) in the second field, or because they have no additional fields. These commands allow the system administrator to optimize archiver operation for the site's configuration and mode of usage.

Specifying an Archive Interval

The archiver executes periodically to examine the status of all mounted ASM file systems. The timing is controlled by the archive-interval time. This time is the time between complete archive operations, scanning and copying to removable media, on each file system. To change the time, use the following command:

`interval =` *time*

The default interval is 10 minutes. If the archiver receives a `SIGALRM` from `sam-init` or from the `arrun` command from within `samu(1M)`, it begins scanning all file systems immediately. `See archiver.cmd(4)`.

Specifying an Archiver Log File

The archiver can produce a log file that contains information about each file archived (or automatically unarchived). The log file is a continuous record of archival action. The file may also be used to locate earlier copies of files for traditional backup purposes. To set the name of the logfile, use the following command:

`logfile =` *filename*

`filename` is the absolute pathname to the log file. By default, this file is not produced.

Controlling Archiving for a Specific File System

By default, archiving controls apply to all file systems. However, the system administrator can confine some controls to an individual file system. To specify an individual file system, use the following command:

`fs =` *`fsname`*

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

Controlling the Number of Drives Used for Archiving

By default, the archiver uses all of the drives in a robot for archiving. To limit the number of drives in a robot used by the archiver, use the following command:

`drives =` *robot count*

Controlling the Size of Archive Files

The maximum size of an archive file is media-dependent. By default, files written to optical disks are no larger than five megabytes. The maximum archive file size for tapes is fifty megabytes. To change the default, use the following command:

`archmax =` *media target-size*

`archmax` may also be set for an individual archive set.

Controlling Volume Overflow

Volume overflow is the process of allowing archived files to span several VSNs. Volume overflow is described in Chapter 2, "ASM File Systems". Before using volume overflow, make sure that you understand the concept. Use volume overflow with caution only after thoroughly assessing the risk. Disaster recovery and recycling are much more difficult with files that span volumes.

Archiver controls volume overflow through the media dependent parameter `ovflmin`. `ovflmin` specifies the minimum size file that is allowed to overflow a VSN. By default, `ovflmin` is set to the capacity of the medium. Thus, only files exceeding the capacity are allowed to overflow a VSN. To change this default, use the following command:

```
ovflmin = media minimum_file_size
```

where *`media`* is the media type and *`minimum_file_size`* is the minimum size of the file to overflow.

For example, many files exist whose length is a significant fraction (say 25%) of the **`mo`** medium. These files partially fill the VSNs and leave unused space on each VSN. To get better packing of the VSNs, set `ovflmin` for the **`mo`** medium to a size slightly smaller than the size of the smallest file, in this instance we set it to 150 Mbytes:

```
ovflmin = mo 150m
```

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

`ovflmin` may also be set for an individual archive set.

Tracing Archiver Activity

The archiver has an extensive action tracing facility. To direct archiver messages to a file, use the following command:

```
trace = filename [event event1 ...]
```

Additional arguments to the command determine which events are traced. This facility is intended to be used for diagnostic purposes. The messages are self-explanatory. See the `archiver.cmd(4)` manual page.

Delaying Archiver Startup

By default, the archiver begins archiving when started by `sam-init(1M)`. `To delay archiving, use the following command:`

```
wait
```

The `wait` command causes the archiver to wait for a SIGUSR1 signal. When this signal is received, normal archiver operations are begun.

Default Directory (datadir)

The archiver `datadir` command allows the site to change the default directory. Set the name of the directory used for archiver's data files to `dirname`. Except for `ReservedVSNs`, the data files are used only during archiver execution.

```
datadir = dirname
```

Delay Time

Indicates the time delay before starting the archiver scan. You can use this time delay to wait for all robots to initialize.

```
delay = time
```
Example:
```
delay = 3m
```

Archive Set Assignments - Assigning Files to Archive Sets

The archive set membership commands assign files with similar characteristics to archive sets. The syntax of these commands is patterned after the `find(1) command.` The syntax of the archive set membership command is:

`archive_set_name` [*`pathname`*] [*`search_criteria`*]

The following are typical archive set membership commands:

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

Archive Set Names

The first field in an archive set assignment is the archive set name. An archive set name is site defined and 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 "_". The first character in the archive set name must be a letter. This also limits family set names. For example, if you use `x-y` as a family set name, the archiver family set fails and archiving does not occur because you used a hyphen (-) instead of an underscore (_).

Path

The path is relative to the mount point of the file system. This allows an archive set membership command to cross-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 this directory, and its subdirectories, are considered for inclusion in this archive set.

File Size

The size of a file may be used to determine archive set membership using the `-minsize and -maxsize` characteristics. The file size may be specified using the suffix letters b, k, M, G, or T for bytes, kilobytes, megabytes, gigabytes, and terabytes. For example,

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

This example specifies 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

The ownership and group affiliation may be used to determine archive set membership using the `-user` and `-group` characteristics. For example,

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

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

Preventing Archival

The archival of files may be prevented by including the files in the archive set `no_archive.` For example,

```
fs = samfs1
no_archive    tmp
```

Files in the `tmp` directory are not archived in this example.

Specifying Filenames Using Regular Expressions

The names of files that are to be included in an archive set may be specified by using regular expressions. The specification `-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(3)` manual page. Note that regular expressions *do not* follow the same conventions as UNIX wildcards.

This example restricts files in the archive set `images` to those files ending with `.gif`:

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

The next example selects files that start with the characters `GEO`:

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

You can use regular expressions with the `no_archive` archive set, as well. For instance, the following prevents any file ending with `.o` from being archived:

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

Setting Release and Stage Attributes

The release and stage attributes associated with files within an archive set can be set using the `-release` and `-stage` characteristics, respectively. Both of these settings override stage or release attributes that may have been set by a user. The following syntax is used to manipulate these file attributes:

```
-release attributes
-stage attributes
```

Attributes for the `-release` parameter follow the same conventions as the `release`(1) command., as follows:

a Release the copy following the completion of the first archive copy

n Never release the copy

p Partially release the file's disk space

Attributes for the `-stage` parameter follow the same conventions as the `stage(1)` command., as follows:

a Associative stage the file

n Never stage the copy

The following example shows how you can use filename specification along and setting file attributes in order to partially release Macintosh resource directories:

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

Archive Set Membership Conflicts

Sometimes the choice of path and other file characteristics for inclusion of a file in an archive set result in ambiguous 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 command file.

When controlling archiving for a specific file system (using the `fs = fsname` command), commands 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`.

Assume, for example, the following `archiver.cmd` segment:

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

At first look 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 segment would be used:

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

Archive Copy Definitions

If you do not specify archive copies, a single archive copy is made for files in the archive set. This copy is made when the archive age of the file is four minutes. If your site requires more than one archive copy, all copies including the first must be specified using the archive copy definitions.

Archive copy definitions are commands that begin with a digit. This digit (1 - 4) is the copy number. These commands are placed immediately after the file characteristics command they apply to. The command syntax is:

copy-number [-release | -norelease] [*archive-age*] [*unarchive-age*]

Releasing Disk Space After Archive

The disk space for files can be automatically released after an archive copy is made by placing `-release` after the copy number.

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

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

Preventing Disk Space Release Until Copies are Completed

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. For instance, 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 off-site vault tapes.

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

Note that the `-norelease` specification on a single copy has no effect on automatic releasing since 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 may be set by specifying the archive-age as the next field on the command. The archive age may be specified with the suffix characters `s`, `m`, `h`, `d`, `w`, or `y` for units of seconds, minutes, hours, days, weeks or years.

```
ex_set data
1 1h
```

In this example, the files in the directory data are archived when their archive age reaches one hour. If the archive age is not specified, the default is 4 minutes.

Automatic Unarchiving

If more than one copy of a file is requested, 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. This example specifies the unarchive age.

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

The first copy of the files in the path `home/users` is archived six minutes after modification. When the files are one week old, a second archive copy is made. The first copy is then unarchived.

More than One Copy for File System Data

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

```
fs = samfs
1 4m
2 1h
```

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

Archive Set Parameters - Controlling the Processing of an Archive Set

This section of the archiver command file is introduced by the `params` command. The section is ended by the `endparams` command.

The parameters for an archive set are defined by lines in the following syntax:

```
archive_set_name.copy_number [[ -param_name
param_value ]...]
```

The sort methods and priority controls can be difficult to manage for a large number of archive sets. The pseudo archive set `allsets` (see `archiver.cmd(4)`) provides a way to set default archive set parameters for all archive sets. All `allsets` commands must occur before those for actual archive set copies. Parameters set for individual archive set copies override parameters set by `allsets` commands.

Assigning Multiple Drives to an Archive Set

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 may be advantageous to use more than one drive. This is specified by using the `-drives` parameter. For example:

```
huge_files.2 -drives 2
```

When the total size of the files in archive set `huge_files.2` exceeds `archmax` for the media, two drives are used to archive the files.

Associative Archiving

When an archive file is written to a medium, files are written to an archive file to efficiently pack the VSN with user files. The user files are sorted by size with the largest files first. Subsequently, when accessing files from the same directory, you may experience delays as the stage process repositions through a VSN 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.

Associative archiving is useful when the contents of files will not change and you wish to access the group of files together at the same time. For example, you might use associative archiving at a hospital for accessing medical images. Images associated with the same patient may 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 rather than the size of the files.

The `-join path` parameter allows these files to be archived contiguously within an archive set copy. For example:

```
patient_images.1 -join path
```

Note: The `-join path` parameter writes data files from the same directory to the same archive file. (An archive file is a batch of files written together in a tar 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 header for each archive file. For most applications, using the `-sort path` parameter is more efficient than using `-join path` if the more restrictive operation of `-join path` is not required.

It is also possible to sort the files within an archive set copy by age or size (the age or size options are mutually exclusive). To sort an archive set, use the `-sort` parameter with the argument `age` or `size`, as follows:

```
cardiac.2 -sort age
catscans.3 -sort size
```

The first example forces the archiver to sort an archive set copy called cardiac.2 by the age of the file, youngest to oldest. The second example forces the archive set copy called catscans to be sorted by the size of the file, largest to smallest.

Splitting Archive Requests among Drives

The `–drivemin min_size` command parameter is used with the `–drives` command to set the minimum size for splitting archive requests among drives. See the manual pages for `archiver.cmd(4)` for more information. (Note that the default for `–drivemin` is `archmax`.)

```
–drivemin min_size
```

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

Use the `-drivemin` parameter with `-drives`. After you set `-drives` and `-drivemin`, an archive request is evaluated against both parameters. If an archive request is less than `drivemin`, only one drive is used to write an archive request. If an archive request is larger than `drivemin`, the archive request is evaluated against `drivemin` and the appropriate number of drives is scheduled. If you do not specify `-drivemin` with `-drives`, the default is `min_size` = `archmax`. To display the setting for `-drivemin`, enter `archiver -lv`.

Use `–drivemin min_size` 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.

In the following 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
`<=10GB`	1
`>10 GB to <= 20 GB`	2
`>20 GB to <= 30 GB`	3
`>30 GB to <= 40 GB`	4
`>40 GB to <= 50 GB`	5
`>50 GB`	5

The following line would be used in the `etc/fs/samfs/archiver.cmd` file for this example:

```
params
bigfiles.1 –drives 5 –drivemin 10G
endparams
```

Reserving VSNs

By default, the archiver writes archive set copies on any medium specified by a regular expression as described in the VSN associations section. However, it is sometimes desirable for archive set VSNs to contain files from only one archive set. Reserving VSNs can be used to satisfy this data storage requirement.

Note: The `-reserve` parameter reserves a VSN for exclusive use by one archive set. A site that uses reserved VSNs probably incurs more media loads and unloads.

The `-reserve` parameter reserves VSNs for an archive set. When set and a VSN has been assigned to an archive set copy, it is not assigned to any other archive set copy, even if a regular expression matches it.

As VSNs are selected for use by an archive set, a ReservedName is assigned to the VSN. `ReservedName` is a unique identifier tying the archive set to the VSN. Each `-reserve` parameter form corresponds to one of three components within the `ReservedName`: the archive set, owner, and file system.

The `-reserve` parameter has three forms. Each form is illustrated below showing the parameter and keyword and examples of reserved names assigned to VSNs. Note that the `ReservedName` syntax consists of the archive set component, pathname component, and file system component, separated by slashes. These slashes are *not* indicative of a pathname; they are merely separators for displaying the three components of a ReserveName. Reserved names are displayed when you use the `archiver -lv` command.

Archive Set Form - The `set` keyword activates the archive set component in the reserved name.

Parameter & Keyword	ReservedName 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.

Parameter & Keyword	ReservedName 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 dir` parameter is intended to reserve a VSN 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 VSN. These small discrete archive files slow the performance of the system because data files are relatively small compared to the tar header for each archive file.

File system Form - The `fs` keyword activates the file system component in the ReservedName.

Parameter & Keyword	ReservedName Examples
`-reserve fs`	`proj.1/p103/samfs1`
	`proj.1/p104/samfs1`

The three forms can be combined and used together in an archive set parameter definition. For instance, the following parameter definition creates a ReservedName based upon an archive set, a group, and the file system:

```
params
tracefiles.1 lt  -reserve set  -reserve group  -reserve
fs
endparams
```

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

```
reserve = set
reserve = [dir|user|group]
reserve = fs
```

A complete archive example using reserved VSNs is presented in this chapter as Example 4. Reserved VSNs are tracked in `/var/adm/samfs/.archive/ReservedVSNs`. See the `ReservedVSNs(4)` manual page for a description of this file.

When the archiver initializes and reserves VSNs to Archive Sets, it reads the `ReservedVSNs` file to determine previous VSN reservations. If this file is not found, a warning message is sent to the syslog.

While reading the `ReservedVSNs` file, archiver edits it to remove badly formatted lines and reservations for relabeled VSNs. A relabeled VSN is identified by a label time that is more recent than the reservation time. Lines referring to previous Archive Set reservations are commented out.

If a reserved VSNs backup file (`/var/adm/samfs/archiver/ReservedVSNs.bak`) does not exist, the current reserved VSNs file becomes the backup. The edited file becomes the new reserved VSNs file. The file consists of ASCII text in the following form:

```
media VSN ArchiveSet/owner/filesystem date time
```

The fields `ArchiveSet`, `owner`, and `file system` may be empty depending on the options in the archiver command file. The date and time indicate when the reservation was made. A reservation line is appended to the file for each VSN that is reserved to an archive set during archiving.

During VSN assignment, the label time of candidate VSNs are compared to the reservation time of the VSN. If the label time is more recent than the reservation time, the reservation is removed and the VSN may be repooled or reused.

Setting Archive Priorities

ASM offers an extensive site controlled priority system for archiving files. Each file is assigned a priority computed from properties of the file and priority multipliers that may be set for each archive set in the `archiver.cmd` file. Properties include online/offline, age, number of copies made, and size (see `archiver(1m)` and `archiver.cmd(4)`). 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.

By setting priorities and sort methods, a site can control the order of file archival. See the following examples for different ways to control the order of file archival.

- 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 VSNs property multiplier
`-priority archive_loaded`	Archive VSN 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 off line 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 VSN loaded property multiplier
`-priority stage_overflow`	Multiple stage VSNs property multiplier

VSN Pools - Naming a Collection of VSNs

A *VSN pool* is a named collection of VSNs. VSN pools are useful for defining a group of media that may be available to an archive set. VSNs are removed from the VSN pool as they are needed for archiving. As such, VSN pools provide a useful buffer for assigning VSNs and reserving VSNs to archive sets.

The system administrator uses VSN pools to define separate groups of VSNs 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 VSNs. A *scratch pool* is a set of VSNs used when specific VSNs in a VSN association are exhausted or another VSN pool is exhausted. (See “VSN Associations - Assigning VSNs to Archive Sets” in the following section.)

As VSNs are selected for use by an archive set they are assigned a “ReserveName” using the archive set component. See the section, “Reserving VSNs for Archive Sets” in this chapter for more information on reserving VSNs. Once a VSN is reserved, it is no longer available to the pool where it originated. Therefore, the number of VSNs within a named pool changes as media are used. The VSN pools can be viewed using the `archiver -lv` command.

The VSN pools section of the archiver command file is introduced by the `vsnpools` command. The section is ended by the `endvsnpools` command or by the end of the `archiver.cmd` file. 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* is to identify the pool. *media_type* is the two character media mnemonic from the `mcf(4)` file as defined in `media(7)`. *vsn_expression* are regular expressions as defined in `regcmp(3G)`.

The following example uses four VSN pools: `users_pool`, `data_pool`, `proj_pool`, and `scratch_pool`. (The pools are defined in the section, “VSN Pools - Naming a Collection of VSNs”.) Note that if one of the three specific pools is out of VSNs, the scratch pool VSNs are selected.

```
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
```

VSN Associations - Assigning VSNs to Archive Sets

This section of the archiver command file is introduced by the `vsns` command. The section is ended by the `endvsns` command.

Collections of VSNs are assigned to archive sets by entries of the following form. An association requires at least three fields: the archive set name and copy, the media type, and at least one VSN. The syntax is as follows:

```
archive_set_name.copy_number  media_type  vsn_expression  ...
```

Note that the archive set name and copy number are connected by the period (.) character. The `media_type` is the media mnemonic as defined in mcf(4).

VSNs are noted by one or more `vsn_expression` which are regular expressions as described in regexp(5). (Note that these regular expressions do not follow the same conventions as wildcards.) When removable media are needed by the archiver for the archive set, each VSN of the selected media in all robots (and manually mounted drives) is examined to determine if it would satisfy any VSN expression. The first VSN that does fit an expression that contains enough space for the archive copy operation is selected.

```
ex_set.1 mo optic[2-3][0-9]
```

This command tells us that files belonging to archive set `ex_set` for copy 1 will be copied to media type mo using any of the twenty VSNs with the name optic20 through optic39

```
ex_set.2  lt  ^TAPE
```

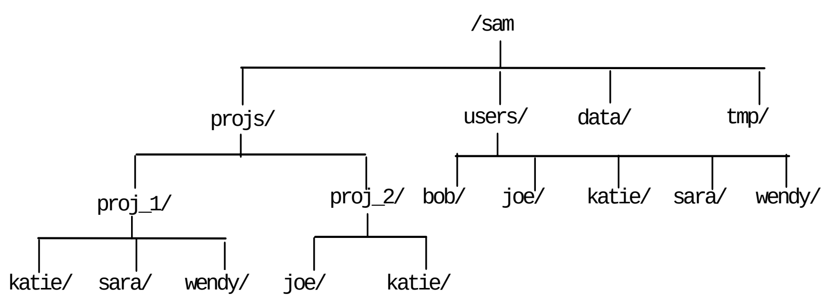
This command copies files belonging to archive set `ex_set` for copy 2 to media type `lt` with any VSN beginning with TAPE.

Note: Make sure you assign VSNs 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.

In addition to a regular expression, you can also specify VSN pools to select VSNs from. Pools are expressed with the "–**pool** ***vsn_pool_name***" parameter with a VSN association.

Directory Structure for all Archiver Examples

All archiver examples assume the following directory structure:



Example 1

This example illustrates the action of the archiver when no archiver command file is used. For this example, there is one file system and an optical robot with two drives and twelve cartridges. The example shows the output produced by the command `archiver -lv.`

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

```
Notify file: /etc/fs/samfs/ar_notify.sh

Archive media:
media:mo archmax: 4.8M Volume overflow not selected
```

The following section indicates the archiver uses five drives. The twelve VSNs, storage capacity, and available space are listed.

```
Archive libraries:
device:hp30 drives_available:5 archive_drives:5
Catalog:
optic00 mo capacity: 622.1M space: 285.4M
optic01 mo capacity: 622.1M space: 138.9M
optic02 mo capacity: 622.1M space: 76.1M
optic03 mo capacity: 622.1M space: 137.5M
optic10 mo capacity: 622.1M space: 218.2M
optic11 mo capacity: 622.1M space: 596.1M
optic12 mo capacity: 622.1M space: 615.7M
optic13 mo capacity: 622.1M space: 218.2M
optic20 mo capacity: 622.1M space: 604.4M
optic21 mo capacity: 622.1M space: 563.9M
optic22 mo capacity: 622.1M space: 621.6M
optic23 mo capacity: 622.1M space: 621.6M
```

The following section shows that both the file system data and data files are included in the archive set `samfs` (the name of the file system). 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
samfs path:
copy:1 arch_age:240
```

The following section shows that files in the archive set `samfs`, copy 1 are archived to all the VSNs in this order.

```
Archive sets:
samfs.1 media: mo (by default)
VSNs:
optic00
optic01
optic02
optic03
optic10
optic11
optic12
optic13
optic20
optic21
optic22
optic23
Total space available: 4.7G
```

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 robot from Example 1. We archive the big files to tape and the small files to optical media.

Here is the archiver command file for Example 2.

```
# Example 2 archiver command file
# Simple selections based on size

logfile = /tmp/archiver.log
interval = 5m

# File selections.
big . -minsize 500k
all .
1 30s

vsns
samfs.1 mo . *0[0-2] # File system data to optic00 -
optic02
all.1 mo .*0 [3-9] .*[1-2][0-9] # All others for files
big.1 lt .*
endvsns
```

All of the following displays show output from the command `archiver -lv example2.cmd.` This section of output is the content of the archiver command file.

```
Reading archiver command file “example2.cmd”
1: # Example 2 archiver command file
2: # Simple selections based on size
3:
4: logfile = /tmp/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.

```
Notify file: /etc/fs/samfs/archiver.sh

Archive media:
media:lt archmax:   47.7M Volume overflow not selected
media:mo archmax:    4.8M Volume overflow not selected

Archive libraries:
device:hp30 drives_available:5 archive_drives:5
  Catalog:
   optic00   mo  capacity:  622.1M  space:  285.4M
   optic01   mo  capacity:  622.1M  space:  138.9M
   optic02   mo  capacity:  622.1M  space:   76.1M
   optic03   mo  capacity:  622.1M  space:  137.5M
   optic10   mo  capacity:  622.1M  space:  218.2M
   optic11   mo  capacity:  622.1M  space:  596.1M
   optic12   mo  capacity:  622.1M  space:  615.7M
   optic13   mo  capacity:  622.1M  space:  218.2M
   optic20   mo  capacity:  622.1M  space:  604.4M
   optic21   mo  capacity:  622.1M  space:  563.9M
   optic22   mo  capacity:  622.1M  space:  621.6M
   optic23   mo  capacity:  622.1M  space:  621.6M
device:lt30 drives_available:5 archive_drives:5
  Catalog:
   TAPE01   lt  capacity:    9.5G  space:    9.5G
   TAPE02   lt  capacity:    9.5G  space:    6.9G
   TAPE03   lt  capacity:    9.5G  space:    4.0G
   TAPE04   lt  capacity:    9.5G  space:    9.5G
   TAPE05   lt  capacity:    9.5G  space:    9.5G
   TAPE06   lt  capacity:    9.5G  space:    8.3G
```

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: /tmp/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:
 all.1 media: mo
  VSNs:
    optic03
    optic10
    optic11
    optic12
    optic13
    optic20
    optic21
    optic22
    optic23
  Total space available:    4.1G

 big.1 media: lt
  VSNs:
    TAPE01
    TAPE02
    TAPE03
    TAPE04
    TAPE05
    TAPE06
  Total space available:   47.6G

 samfs.1 media: mo
  VSNs:
    optic00
    optic01
    optic02
  Total space available:  500.4M
```

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 fifteen-minute interval and an archival record kept. The example shows the output from the `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 = /usr/tmp/archive.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
```

```
Notify file: /etc/fs/samfs/ar_notify.sh

 Archive media:
 media:lt archmax:   47.7M Volume overflow not selected
 media:mo archmax:    4.8M Volume overflow not selected

 Archive libraries:
 device:hp30 drives_available:5 archive_drives:5
   Catalog:
    optic00   mo  capacity:  622.1M  space:  285.4M
    optic01   mo  capacity:  622.1M  space:  138.9M
    optic02   mo  capacity:  622.1M  space:   76.1M
    optic03   mo  capacity:  622.1M  space:  137.5M
    optic10   mo  capacity:  622.1M  space:  218.2M
    optic11   mo  capacity:  622.1M  space:  596.1M
    optic12   mo  capacity:  622.1M  space:  615.7M
    optic13   mo  capacity:  622.1M  space:  218.2M
    optic20   mo  capacity:  622.1M  space:  604.4M
    optic21   mo  capacity:  622.1M  space:  563.9M
    optic22   mo  capacity:  622.1M  space:  621.6M
    optic23   mo  capacity:  622.1M  space:  621.6M
 device:ml40 drives_available:5 archive_drives:5
   Catalog:
    TAPE01   lt  capacity:    9.5G  space:    9.5G
    TAPE02   lt  capacity:    9.5G  space:    6.9G
    TAPE03   lt  capacity:    9.5G  space:    4.0G
    TAPE04   lt  capacity:    9.5G  space:    9.5G
    TAPE05   lt  capacity:    9.5G  space:    9.5G
    TAPE06   lt  capacity:    9.5G  space:    8.3G

 Archive file selections:
 Filesystem samfs  Logfile: /usr/tmp/archive.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:106
     copy:1  arch_age:60
     copy:2  arch_age:60
 pict  path:. gid:10001
     copy:1  arch_age:60
     copy:2  arch_age:60
 no_archive Noarchive path:tmp
 samfs  path:.
     copy:1  arch_age:240
```

```
Archive sets:
joe.1 media: mo
VSNs:
optic01
Total space available: 138.9M

joe.2 media: lt
VSNs:
TAPE01
TAPE02
Total space available: 16.4G

pict.1 media: mo
VSNs:
optic02
Total space available: 76.1M

pict.2 media: mo
VSNs:
optic03
Total space available: 137.5M

prod.1 media: mo
VSNs:
optic20
optic21
optic22
optic23
Total space available: 2.4G

prod_big.1 media: lt drives: 2
VSNs:
TAPE03
TAPE04
Total space available: 13.5G

prod_big.2 media: lt drives: 2
VSNs:
TAPE05
TAPE06
Total space available: 17.7G

proj_1.1 media: mo
VSNs:
optic10
optic11
Total space available: 814.2M
```

Continued from the previous output example:

```
proj_1.2 media: mo
VSNs:
optic12
optic13
Total space available: 833.9M

samfs.1 media: mo
VSNs:
optic00
optic01
Total space available: 424.3M
```

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 VSNs. This example shows how to reserve VSNs 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 example shows the `archiver.cmd` file and the output from the `archiver -lv` after the VSNs have been reserved by a pass of the archiver.

```
logfile = /var/adm/archiver.log
interval = 6000

users users
1 30s

data data
1 30s

proj projects
1 30s

params
users.1 -reserve user
data.1 -reserve group
proj.1 -reserve dir -reserve fs
endparams

vsnpools
users_pool mo OPT0[2-4][a-b]
data_pool mo OPT0[5-7][a-b]
proj_pool mo OPT0[8-9][a-b]
scratch_pool mo B.*
endvsnpools

vsns
samfs1.1 mo OPT00[a-b]
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
```

```
Reading archiver command file "archiver.cmd"
Notify file: /etc/fs/samfs/ar_notify.sh
Archive media:
media:dt archmax:   47.7M Volume overflow not selected
media:mo archmax:    4.8M Volume overflow not selected
Archive libraries:
device:pm30 drives_available:2 archive_drives:2
  Catalog:
   OPT00a   mo  capacity: 1.2G  space: 1.2G  reserved:samfs1.1//
   OPT00b   mo  capacity: 1.2G  space: 1.2G
   OPT01a   mo  capacity: 1.2G  space: 1.2G  reserved:samfs2.1//
   OPT01b   mo  capacity: 1.2G  space: 1.2G
   OPT02a   mo  capacity: 1.2G  space: 1.2G  reserved: users.1/user5/
   OPT02b   mo  capacity: 1.2G  space: 1.2G  reserved: users.1/user4/
   OPT03a   mo  capacity: 1.2G  space: 1.2G  reserved: users.1/user3/
   OPT03b   mo  capacity: 1.2G  space: 1.2G  reserved: users.1/user2/
   OPT04a   mo  capacity: 1.2G  space: 1.2G  reserved: users.1/user1/
   OPT04b   mo  capacity: 1.2G  space: 1.2G
   OPT05a   mo  capacity: 1.2G  space: 1.2G  reserved:
data.1/engineering/
   OPT05b   mo  capacity: 1.2G  space: 1.2G  reserved:
data.1/chemistry/
   OPT06a   mo  capacity: 1.2G  space: 1.2G reserved: data.1/biology/
   OPT06b   mo  capacity: 1.2G  space: 1.2G reserved:data.1/astronomy/
   OPT07a   mo  capacity: 1.2G  space: 1.2G
   OPT07b   mo  capacity: 1.2G  space: 1.2G
   OPT08a   mo  capacity: 1.2G  space: 1.2G reserved:
proj.1/p105/samfs1
   OPT08b   mo  capacity  1.2G  space  1.2G reserved:
proj.1/p104/samfs1
   OPT09a   mo  capacity: 1.2G  spac   1.2G reserved:
proj.1/p103/samfs1
   OPT09b   mo  capacity:  .2G  space: 1.2G reserved:
proj.1/p102/samfs1
device:dt40 drives_available:1 archive_drives:1
  Catalog:
   TAPE   dt  capacity:   1.9G  space:    1.9G
device:rb1 drives_available:5 archive_drives:5
  Catalog:
   B02009   mo  capacity: 97.7M space: 97.6M  reserved:
proj.1/p101/samfs1
   B02010   mo  capacity: 97.7M space: 97.6M  reserved:
proj.1/p104/samfs2
   B02011   mo  capacity: 97.7M space: 97.6M  reserved:
proj.1/p103/samfs2
   B02012   mo  capacity:   97.7M  space:   97.6M
   B02013   mo  capacity:   97.7M  space:   97.6M
   B02014   mo  capacity:   97.7M  space:   97.6M
   B02015   mo  capacity:   97.7M  space:   97.6M
   B02016   mo  capacity:   97.7M  space:   97.6M
   B02017   mo  capacity:   97.7M  space:   97.6M
   B02018   mo  capacity:   97.7M  space:   97.6M
   B02019   mo  capacity:   97.7M  space:   97.6M
```

```
Archive file selections:
Filesystem samfs1  Logfile: /var/adm/archiver.log
samfs1 File system data
    copy:1  arch_age:240
users  path:users
    copy:1  arch_age:30
data  path:data
    copy:1  arch_age:30
proj  path:projects
    copy:1  arch_age:30
samfs1  path:.
    copy:1  arch_age:240

Filesystem samfs2  Logfile: /var/adm/archiver.log
samfs2 File system data
    copy:1  arch_age:240
users  path:users
    copy:1  arch_age:30
data  path:data
    copy:1  arch_age:30
proj  path:projects
    copy:1  arch_age:30
samfs2  path:.
    copy:1  arch_age:240

VSN pools:
data_pool media: mo VSNs:
   OPT07a
   OPT07b
 Total space available:    2.4G

proj_pool media: mo  ** No VSNs available. **

scratch_pool media: mo VSNs:
   B02012
   B02013
   B02014
   B02015
   B02016
   B02017
   B02018
   B02019
 Total space available:  780.9M

users_pool media: mo VSNs:
   OPT04b
 Total space available:    1.2G

Archive sets:
data.1 media: mo reserve:set/group/
 VSNs:
   OPT07a
   OPT07b
   B02012
```

```
      B02013
      B02014
      B02015
      B02016
      B02017
      B02018
      B02019
  Total space available:    3.2G

proj.1 media: mo reserve:set/dir/fs
  VSNs:
      B02012
      B02013
      B02014
      B02015
      B02016
      B02017
      B02018
      B02019
  Total space available:  780.9M

samfs1.1 media: mo reserve:set//
  VSNs:
      OPT00a
      OPT00b
  Total space available:    2.4G

samfs2.1 media: mo reserve:set//
  VSNs:
      OPT01a
      OPT01b
  Total space available:    2.4G
users.1 media: mo reserve:set/user/
  VSNs:
      OPT04b
      B02012
      B02013
      B02014
      B02015
      B02016
      B02017
      B02018
      B02019
  Total space available:    2.0G
```

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 ASM and the archiver, and ensure that your data is stored in the safest way possible.

Each site is unique in its application of computing and data storage hardware and software. The following recommendations are based upon our experiences using ASM with real world applications. When writing the archiver commands file for your site, be sure that you reflect the data storage requirements at your site.

1. *Save your archive logs.* The archive logs provide information that is essential to recovering data, even when ASM is unavailable. It is recommended that you keep these logs in a safe place in the event of a catastrophic disaster during which ASM is unavailable.

2. *Use regular expressions for VSNs*. Let the system work for you by allowing it to put files on many different VSNs. Volume ranges (specified using regular expressions) allow the system to run continuously. Using specific VSN names for archive set copies can rapidly fill a VSN, 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 increase the performance of the archiver compared to one ASM file system. The archiver is a multi-threaded 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 systems (like UNIX file systems). For performance considerations, we recommend that you do not place more than 1000 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 medium 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, etc.) 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.

Troubleshooting the Archiver

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

- `samu(1M)` "a" display- This display shows archiver activity for each file system. It also displays archiver errors and warning messages, such as " *** 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 media. Archive logs can become excessively large and should be reduced regularly by a `cron(1)` job or by the system administrator. Archive these log files for safekeeping, since 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. "`sls -D`" displays inode information for a file, including copy information.

Note: "`sls -D`" might display 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. An archive copy exists only when you can view the copy information displayed by `sls`.

Why Files Are Not Archiving

Here is a checklist of reasons why ASM might not be archiving files:

- `archiver.cmd has a syntax error. Run "archiver -lv"` to identify the error, then correct the flagged lines.
- `archiver.cmd` has a "wait" command in it. Either remove the "wait" command or override it by from `samu(1M)` by entering **`:arrun`**.
- No media is available. This is also shown with "`archiver -lv".` Add more media as needed. You may have to export existing media to free up slots in the library.

- The tapes for an archive set are full. You can export tapes and replace them with new media (make sure the new media is labeled), or recycle the media (see Chapter 8, "The Recycler").
- `archiver.cmd` VSN section fails to list correct tapes. Check your regular expressions and VSN pools to ensure they are correctly defined.
- There is not enough space to archive any file on the available media. If you have larger files and it appears that the media are nearly full, the media may be as full as ASM allows. If this is the case, add media or recycle as described above.
- `archiver.cmd` has the "`no_archive`" command set for directories or file systems that contain large files.
- "`archiver -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 problems with the robot.

Additional Archiver Diagnostics

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

1. Check the syslog file (default is `/var/adm/sam-log)` for any archiver messages that may 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 media 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 to determine the system call that is not responding.

Why Files Are Not Releasing

The `archiver(1M)` and `releaser(1M)` 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. See Chapter 7, "The Releaser" , for more information on why files are not being released. Some of the common reasons that ASM is not releasing files are listed below:

- Files can only be released after they are archived. Check "Why Files Are Not Archiving" above.
- `archiver.cmd` has the "`-norelease`" parameter set for too many archive copies.

- The “`release -n`” command has been used to flag directories and files as “never release”.
- The `archiver.cmd` has the “-release n” set for too many directories and files.
- The `releaser(1M)` high water mark is set too high and automatic releasing occurs too late. Verify this in the `samu “m”` display or with libmgr(1M) and lower it.
- The `releaser(1M)` low water mark is set too high and automatic releasing stops too soon. Check this in the `samu “m”` display or with libmgr(1M) and lower it.
- Large files are busy. They will never reach their archive age and be archived and released.

Chapter 7. The Releaser

Introduction

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 file system invokes the releaser process when a site-specified disk threshold is reached. In contrast, the `release` command allows users to immediately release a file's disk space or set releasing parameters for a file.

Releaser Overview

When the cache utilization for a file system exceeds its configured high water mark, the file system sends a message to `sam-init` to invoke the releaser. First, the releaser reads its command file and collects the parameters that control the release process. Next, it scans the file system and collects information about each file. 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.

Definitions

The following definitions clarify the information presented in this chapter.

Age The amount of elapsed time from a given event until now. A file's `inode` keeps track of the times that the releaser uses (see the following list).

- residence change time
- data modified time
- data accessed time

You can view these times using `sls –D` (see `sls(1)` manual page). 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 five minutes.

Candidate A file eligible to be released. See the following list of reasons why a file would not be a candidate.

- The file is already off-line.
- 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.
- 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 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 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 Command File

Several weights are used to determine the release priority of a file. In ASM versions prior to 3.3.1, these parameters could only be specified on the mount command or in the options field in the `/etc/vfstab` file. These mount options are now replaced by the `/etc/fs/samfs/releaser.cmd` file. Future versions of the software may remove the mount options.

See the `releaser.cmd(4)` manual page for more details. The `releaser.cmd` file consists of lines of text that are either comment lines denoted by a leading # or command lines. The commands perform the following function:

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

Caution: Make sure there are spaces on both sides of all equal signs (" = ") 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 ASM.

Releaser Log File

If the command `logfile = <filename>` is specified in the `releaser.cmd` file, the releaser appends its activity to the indicated filename, creating the filename if it does not exist. Here is a sample log file:

```
Releaser begins at Wed Apr 28 17:29:06 1999
inode pathname          /sam1/.inodes
low-water mark          24%
weight_size             1
weight_age              1
fs equipment ordinal    1
family-set name         samfs1
started by sam-init?    yes
release files?          yes
display_all_candidates? no
---before scan---
blocks_now_free:     3481504
lwm_blocks:          3729362
---scanning---
10501 (R: Wed Apr 21 18:47:50 CDT 1999) 10001 min, 500
blks /sam1/testdir0/filevp
10500 (R: Wed Apr 21 18:48:10 CDT 1999) 10000 min, 500
blks /sam1/testdir0/filewq
...
---after scan---
blocks_now_free:       3730736
lwm_blocks:            3729362
archnodrop: 0
already_offline: 0
bad_inode_number: 0
damaged: 0
extension_inode: 0
negative_age: 0
nodrop: 1
not_regular: 9
number_in_list: 675
released_files: 202
too_new_residence_time: 0
too_small: 2
total_candidates: 675
total_inodes: 1376
wrong_inode_number: 0
zero_arch_status: 689
zero_inode_number: 0
zero_mode: 0
CPU time: 2 seconds.
Elapsed time: 10 seconds.
Releaser ends at Wed Apr 28 17:29:16 1999
```

The `releaser(1M)` manual page describes the information contained in the log file. Since 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 "---after scan---":

```
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

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 following commands specify the priority weights used to calculate the release priority for a file. The default value for weights is 1.0.

```
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.

```
release-priority  = age-related-priority + size-related-priority
```

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

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.

Because specifying the priority for the age of a file is a complex process, two sets of parameters are available.

`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 `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, modify, 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 kbyte blocks) is multiplied by a factor called `weight_size` to obtain the size-related portion of the priority.

Specifying Differing Parameters for File Systems

Use the `fs = family_set_name` command in the command file to indicate that the commands following "`fs = `" apply only to the given file system. Commands preceding the first "`fs = `" command are global and apply to all file systems. Commands following "`fs = `" override global commands.

The `releaser.cmd(4)` manual page includes examples of the "`fs = `" command.

Releaser Operation

A file system may contain thousands of files. Keeping track of the release priority for all the files can be wasteful, since 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.

Once 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 (perhaps because all remaining on-line files are marked "`release -n`"). ASM starts the releaser again later.

How to Configure Releaser for Files in Cache

It is necessary to decide the characteristics of files in cache for your site. Since it is wasteful to mount a tape if you are only staging a few kilobytes, you may want to bias your system to retain small files in cache. In this case, make `weight_size 1` and `weight_age 0`. This causes the releaser to release files in strict largest first order.

The `releaser.cmd` parameters that release the largest files first are listed below:

```
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. Set `weight_age_modify` to `1.0`, `weight_age_access` to `0.0, weight_age_residence` to `0.0`, `and weight_size` to `0.0`. This set of age weights releases files in strict order starting with the oldest modified to the most recently modified.

The following list includes the `releaser.cmd` parameters that release the oldest modified files first.

```
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: A system administrator wants to release the largest files first. There are hundreds of small files that are the same size, and 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` is zero, the releaser is then forced to release the small files in essentially random order because they are all the same size and with the same release priority.

In this scenario, you could set `weight_age` to `0.01` as a tiebreaker. The releaser would then 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: Set the `weight_size = 1.0` and the `weight_age = 0.01`.

This method violates 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 above settings, a 4 kbyte file that staged 100 minutes ago and an 8 kbyte file that just staged both have the same release priority.

The releaser randomly chooses a file to release. If it chooses a 4 kbyte file, it violates the administrator's largest-first intent. Setting `weight_age` considerably smaller (e.g., `0.001)`, reduces this effect. If a 4 kbyte file staged 1,000 minutes ago, it has the same priority as the 8 kbyte file that just staged.

You can use the `no_release` and `display_all_candidates` commands, and run the releaser manually to obtain a list of candidates in priority order for use in adjusting the priority weights.

Running 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.

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

```
/etc/fs/samfs/releaser x /sam1 47 1.0
```

The "`x`" indicates compatibility with previous versions of ASM 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 displays information on the terminal and in the releaser log file (if specified in the command file.)

Troubleshooting the Releaser

There are several reasons why the releaser cannot release a file.

- 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 when:
  - the archiver has just staged an offline file to make an additional copy.
  - the –`norelease` attribute 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 8. The Recycler

Introduction

As archiving progresses within the ASM system, unused archive copies can occupy an increasing amount of media space within a media library. The recycler process reclaims the space occupied by unused archive copies. This chapter presents an overview, theory of operations, and configuration of the recycler as part of the ASM automated storage management system.

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 medium 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 medium starts out with all its capacity as free space. As data is archived to the medium, 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 become stale 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 stale 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.

Process of Recycling

Because tape media only can be appended to and cannot be re-written in place, the only way to recycle is to move all the current data off a medium, relabel the medium, and start using it again from the beginning. To achieve this, the archiver identifies all the archive images present on a VSN. It marks these images to enable the archiver to replace the copy on the VSN being recycled with a copy on another VSN. This operation is called "rearchiving". 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 medium.

Prior to the 3.3.0 release of ASM, recycling was triggered when the media utilization in an entire media changer exceeded the media changer's configured high-water mark. With release 3.3.0 of ASM, recycling may be triggered either by media changer or by archive set utilization.

Recycler 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 media changer catalogs and the inodes.

Each time it is run, the recycler performs these steps:

1. Verifies that no other instances of the recycler are currently running.
2. Reads the archiver's "archset" file.
3. Scans the media changer catalogs.
4. Reads the `recycler.cmd` file.
5. Reads the archiver's "reserved VSN" file.
6. Verifies that all ASM file systems are mounted.
7. Scans each file system's `.inodes` file and builds the VSN table.
8. Assigns appropriate VSNs to the appropriate archive sets.

9. Sorts the VSN table.
10. Stops running immediately if errors are detected.
11. Displays the catalogs. These include both the physical media changer catalogs and the pseudo-catalogs created for each archive set.
12. Selects candidate VSNs for each media changer and archive set being recycled.
13. Displays the VSN table.
14. Marks archive copies on recycling VSNs "to be rearchived".
15. Runs `recycler.sh` for all VSNs that are recycling and have no archive copies.

1. Verify a single instance of the recycler running

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

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 parameters section of the `archiver.cmd file` with one or more recycling parameters (bracketed by the keywords `param/endparams`). These archive sets are called *recycling archive sets*. See the `archiver.cmd(4)` manual page for additional information.

3. Scan the media changer catalogs

The recycler obtains the catalog file for each media changer configured in the system's `mcf` file. Note that media in manually mounted drives cannot be recycled.

The catalog contains the media type, VSN, space and capacity values for each VSN in the media changer.

Space and capacity are reported by the drive after each file is written to a medium. These values are based on the drive's sense of how much of the medium has physically been used, and on its knowledge of the recording density on the medium. 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 the medium is full; capacity is the total amount of data that fits on the medium. A 10 Gbyte tape with 3 Gbytes written to it already has a capacity of 10 Gbytes, and a space of 7 Gbytes.

4. Read the `recycler.cmd` file

The recycler reads and checks the `/etc/fs/samfs/recycler.cmd` file. See the `recycler.cmd(4)` manual page for a complete description of the command options and syntax. This file contains options for processing each media changer in the system.

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

```
logfile = /usr/tmp/recycler.log

stk30  51  60  ignore  mail  root
```

The options are the high-water mark, the minimum VSN gain, an address to which mail concerning this media changer should be sent, and the word "ignore" indicating that recycling should not occur for this media changer.

The high-water mark allows the administrator to set the percentage of media usage below which recycling not occur. This percentage is the ratio of the used space in the media library to its total capacity. For this example, a media changer that holds 10 20-Gbyte tapes, three of them 100% full and the remaining seven each 30% full, would have a media utilization of: ((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 media changer's VSNs for recycling.

Note: You can force a VSN to be recycled by using the `chmed +c` command to set the recycling flag. See the `chmed(1M)` manual page for additional information.

The minimum VSN-gain percentage is used to set a lower limit on the amount of space to be gained by recycling a medium. For example, if a medium in a media changer is 95% current data and 5% expired data, the gain obtained by recycling the medium 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 medium with 90% expired data, 5% current data, and 5% free space that would have a gain of 90% if recycled.

The mail option tells the recycler to send mail when recycling occurs on a given media library. The mail message has a subject line of "Robot *robot-name* recycle." Sample message bodies could be:

> 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 media changer.

5. Read the archiver reserved VSN file

The archiver keeps a reserved VSN file (`ReservedVSNs`) that contains VSNs dedicated to a specific archive set. The recycler reads this file to obtain the list.

6. Verify all ASM file systems are mounted

The inodes contain information about the volumes used to archive a given file. Therefore, all configured file systems must be mounted to allow the `.inode`s 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 medium.

Note that this check is not foolproof. For example, assume that you have just completed a `samfsdump, sammkfs`, and `mount` on a file system with the intent of `samfsrestore-ing` 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. It is vitally important that the administrator disable the recycler whenever such situations occur.

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.

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 media-changer catalogs, and the file systems' `.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 media changer from which they came, to be recycled by that media changer.

9. Sort the VSN table

The VSN table is augmented with the space and capacity values from the catalog. The entire VSN table is sorted to first group VSNs by media changer 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.

10. Stop running now if any errors were detected

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

11. Display the catalogs

For each media changer and recycling archive set, and 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.

12. Select a candidate for each media changer being recycled

For each media changer and archive set that needs recycling (its media utilization is larger than its high-water mark) and is not ignoring recycling, find a candidate VSN to recycle. If any VSN in the media changer 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. Mark that VSN "recycling".

The recycling flag is stored in the catalog and displayed in `samu` and the GUI's. 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.

13. Display the VSN table

The VSN table is displayed. See the `recycler(1M)` manual page for more information.

14. Mark archive copies on recycling VSNs "to be rearchived"

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

15. Run recycler.sh

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.

The `/etc/fs/samfs/recycler.sh script` is shipped by LSC, Inc. configured to send mail to root when invoked. If you wish, you can modify the script to automatically relabel media that has finished recycling. The following example shows a portion of a `recycler.sh` script that relabels media:

```
/opt/LSCsamfs/sbin/chmed -R mo $2
/opt/LSCsamfs/sbin/${1}label -vsn $2 -old $2 -slot $3 $4
```

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 command.
- Do not run the recycler while performing maintenance on an ASM 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.
- The recycler must be run when all ASM file systems are mounted.
- The recycler is started by running the `recycler(1M)` command from the command line or from a `cron(1)` job.

Step 1: Setup the recycler.cmd File

Write the `/etc/fs/samfs/recycler.cmd` file as specified in `recycler.cmd(4)`. Determine a location for the log file. Select a percentage for each robot's high water mark and VSN minimum-percent-gain and specify these entries `in /etc/fs/samfs/recycler.cmd. 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 using recycle by archive set, you still should configure each media changer in the `recycler.cmd` file. This ensures that even VSNs that do not fall into an archive set can be recycled if needed.

Step 2: Edit the archiver.cmd file (recycle by archive set only)

Edit `/etc/fs/samfs/archiver.cmd` and add similar information for the archive sets you wish to recycle. As noted above, include the `-recycle_ignore` parameter to prevent the recycler from taking action before your configuration has been tested.

Step 3: Run recycler

Run `/opt/LSCsamfs/sbin/recycler.` Examine the standard output, log, sam-log and `/var/adm/messages` for any error messages from the recycler. A recycler log follows.

```
========== Recycler Wed Jul  2 14:30:06 1997 ===========
1 catalog:
0  Family: hp30                  Path: /etc/fs/samfs/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 things seem okay, 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 commands

Remove the `-recycle_ignore` and `ignore` parameters 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 following example shows how to relabel a recycled VSN and send mail to the super user.

```
#!/bin/csh
#
#   /etc/fs/samfs/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" - use to construct the name
#           of the appropriate label command: odlabel or tplabel
#
#      $2 - VSN being post-processed
#
#      $3 - slot number in the library where the VSN is located
#
#      $4 - equipment number of the library where the VSN is located
#
#
# $Id: recycler.sh,v 1.2 1997/03/05 22:38:06 jlh Dev $
#
#   As an example, if uncommented, the following line relabels the
VSN:
#
/opt/LSCsamfs/sbin/chmed -R mo $2
/opt/LSCsamfs/sbin/${1}label -vsn $2 -old $2 -slot $3 $4
#exit
#   These lines 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
#  The default action is to mail a message reminding you to set up this
#  file. Comment out these lines after you've set up this file.
#
mail root <</eof
```

The recycler called the `/etc/fs/samfs/recycler.sh` script with the following arguments:

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

`/etc/fs/samfs/recycler.sh` is a script 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 choose to relabel and reuse the media; others choose to remove the media from the library to use later for accessing historical files. Consult the `recycler(1m)` manual page for more information.

```
/eof
```

Troubleshooting the Recycler

The most frequent problem is the following message displayed by the recycler whenever 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:

- The archiver fails to re-archive the 123 archive copies on the volume because:
  - Files that need to be re-archived are marked "no archive".
  - Files that need to be re-archived are in the `no_archive` archive set.
  - Files cannot be archived because there are no available VSNs.
  - `archiver.cmd` file contains a `wait` statement.

OR

- 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. Since the inodes are not part of the directory tree, they are not accessible by a file name. The archive copies are not active.

To determine which condition is true, run the recycler with the "`-v`" option. This displays the pathnames of the files associated with the 123 archive copies in the recycler log file:

> 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, seven pathnames are displayed along with one message, `"Cannot find pathname..."`. To correct the problem with LSDAT1 not draining, you need to determine why the seven files cannot be re-archived. After the seven files are re-archived, 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 finding the pathname, 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 medium 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 recycler selects VSNs for recycling.

Chapter 9. Dumping & Restoring ASM Control Structures

Introduction

This chapter describes the procedures for dumping up and restoring ASM file systems control structures. The following topics are covered in this chapter.

- An overview of ASM control structures
- How to dump and restore file systems using `samfsdump`/`samfsrestore`
- How to dump and restore file systems using the `.inodes` file

An Overview of ASM Control Structures

This overview describes the ASM control structure information and introduces two methods used to dump and restore ASM file systems. Control structures, file system design, and backup methodologies are presented to help you determine the appropriate dump/restore for your site.

ASM Control Structures

File systems consist of directories, files, and links. ASM maintains file systems by keeping track of all files in the `.inodes` file. The `archiver(1M)` stores the data portion of files on removable media. Directory, link, and inode information, which makes up the control structure, is just as important as the data in the event of a disaster. Restoring this metadata allows you to bring up the file system quickly, providing access to data which is now offline.

The `archiver(1M)` archives the directories and links for a file system in an archive set that matches the file system name. For example, if you created a disk cache file system with a family set name of samfs1, there is also an archive set called samfs1. This archive set encompasses the directory and link information and any files that are not included in another archive set.

Note, however, that the `.inodes` file is not archived. In the event of a disaster, the control structure needs to be restored before the data can be accessed. The control structure identifies where the data resides in the file system.

It is important to periodically perform control structure dumps to protect your data from a disaster. The dumps should be done at least once a day, but the frequency is dependent on your site's requirements. By dumping control structures (using `samfsdump(1M)`) on a regular basis old files can be restored (using `samfsrestore(1M)`) even if they have been removed from the file system, as long as the media has not been erased. Also, by saving the control structure dump and the archive tapes, files or entire file systems may be moved from one file system to another, or even from one server to another.

ASM File System Design

It is important to design your file system so that you can quickly access information as well as recover information when needed.

ASM is a multi-threaded advanced storage management system. To take advantage of these capabilities, you should create multiple file systems whenever possible.

In ASM file systems, like Solaris file systems, directory lookups use a linear search method, searching from the beginning of the file system to the end. As the number of files in a directory increases, the search time through the directory increases. Users who have directories with over 1000 files experience excessive search times.

These search times are also evident when you restore a file system. To increase performance and speed up file system dumps and restores, you should keep the number of files in a directory under 1000.

Making Control Structure Backups for File Systems

There are two ways to dump and restore the control structure information for file systems: performing a `samfsdump` and `samfsrestore`, or by copying the `.inodes` file and restoring using `sammkfs`. Both methods have advantages and disadvantages.

`samfsdump/samfsrestore` creates and restores control structure dumps of a current directory. `samfsdump` 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 for restoring the control structure for a single directory or file. `samfsdumps` are performed on mounted file systems and run relatively quickly at about 100,000 files per minute, depending on the layout of your file system.

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 generally faster than a `samfsdump`, however, there is a downside. No directory, link, or file path information is saved when copying the `.inodes` file, so directory information (previously saved in the default file system archive sets, described in “Control Structures” above) is staged automatically when restoring the file system. This can be time consuming depending on the location of the archive set on the medium. Also, you cannot restore a portion of a file system like you can with `samfsrestore`.

Given these attributes for dumping and restoring file systems, StorageTek recommends performing full `samfsdump` backups once or twice a day with backups of the `.inodes` file more frequently, perhaps once each hour. You need to determine which method and frequency is right for your site.

How To Dump and Restore File System Control Structures

The following are general guidelines for performing dumps:

- ASM dumps are performed with the file system mounted. Inconsistencies may arise between stale archive copies and newly created files (a *stale* archive copy refers to the situation when a file is updated and a new archive image has not been created yet. Prior to the new image being created, the old image is marked stale). Dumping control structures during a quiet period (a time when files are not be created or modified) is a good idea and may eliminate these inconsistencies.
- Perform dumps on a regular basis. You can run `samfsdump` 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`s and copies of the .inodes file in order to guard your system from such a failure.
- Be aware that dumping control structures when there are unarchived files present results in a warning message indicating that these files will be damaged. This message indicates the file will not be available when this dump is restored.
- Ensure that you dump control structures for all ASM file systems. Look in `/etc/vfstab` for all file systems of type `samfs`.

How to Dump Control Structures Using samfsdump/samfsrestore

You can use the `samfsdump(1M)` command to dump control structures information; the `samfsrestore(1M)` command is used to restore control structures generated by `samfsdump.` See the manual page for a complete description of the syntax and options available with these commands.

To manually create an ASM dump for a file system, or a directory within a file system, enter the following commands:

1. Login as root.

2. Change to the mount point for the file system or to the directory.

   ```
   server# cd /sam1
   ```

3. Create a dump file by executing the `samfsdump` command.

   ```
   server# samfsdump -f dump_file
   ```

To automate your dump procedures, use `cron(1)` to create an entry in root's cron table. The following example entry performs a dump and manages the files within a dump directory. Note that the xargs argument is 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 sam1; /
/opt/LSCsamfs/sbin/samfsdump -f /
/csd.directory/sam1/'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 midnight. First, dump files older than three days are removed and a new dump is created in `/csd.directory/sam1/`*`yymmdd.`*

If you have multiple ASM file systems, make similar entries for each. Be sure that you save each dump in a separate file.

How to Restore File Systems

This example assumes that you have an ASM dump file *`dump_file`* as illustrated in the previous subsection.

1. Change into an ASM file system and create a new directory.

   ```
   server# cd /sam1
   server# mkdir restored_fs
   server# cd restored_fs
   ```

2. Restore the directory using an existing dump file.

   ```
   server# samfsrestore -f dump_file
   ```

How to Restore Single Files and Directories

You can also use a `samfsdump` file to restore the control information for a single file or directory, relative to the current directory. Enter the following:

1. List the name of the file or directory that you want restored.

   server# **samfsrestore –t -f *dump_file***

2. Restore the file relative to the current directory. ***file_name*** must exactly match the name of the file or directory as it was listed in the step above.

   server# **samfsrestore -f *dump_file* *file_name***

Error Messages

When ASM files are modified, the archive copies are marked stale (this is seen in the output of the `sls -D` command). This causes new archive copies to be made. If a `samfsdump` is performed prior to the creation of the new archive copies, the following warning message is displayed when `samfsdump` is run:

/path/to/the/file: Warning! File will be damaged

This warning indicates that the file has not yet been archived. Upon restoring the file with `samfsrestore`, the following message appears indicating that the file is now damaged:

/path/to/the/file: Warning! File is now damaged

How to Dump Control Structures Using .inodes

For backup purposes, you can manually create a copy of the .inodes file as follows:

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**

To automate the copy of the .inodes file use `cron(1)` to create an entry in root'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 * * * ( rm -f /var/sam/samfs1.inodes.old;
mv -f /var/sam/samfs1.inodes /var/sam/samfs1.inodes.old;
/bin/dd if=/sam/.inodes of=/var/sam/samfs1.inodes bs=512k)
```

In the above examples, note that you can use a variable block size when using the `dd` command. Block sizes must be multiples of 16k, for instance bs=16k, bs=128k, etc. An increased block size increases performance of the copy.

How To Restore A File System from .inodes

The file system is recreated using the copied `.inodes` file as input to `sammkfs`. `sammkfs` constructs the file system and restores the inodes. All the files, directories, symbolic links, and removable media files are off-line. In order for the file system to be completely restored, you (or the Archiver) must traverse the directory structure, which causes this directory to be staged in.

1. Restore the file system using the .inodes file as input to `sammkfs.`

   server# **sammkfs -r /var/sam/samfs1.inodes samfs1**

2. Mount the file system.

   server# **mount samfs1**

3. Change directories to the ASM mount point and traverse the directory structure, either by allowing the archiver to scan the file system or by entering `sfind` as shown here.

   server# **cd /*sam***

   server# **sfind . -type d > /dev/null**

Chapter 10. Windows-Based Administration Tools

Introduction

This chapter describes the graphical user interfaces (GUIs) used to manage ASM. There are two GUIs used to manage robots, devices, and media mount requests:

- `libmgr(1M)` - This Java-based, all-purpose GUI will replace `samtool` in the near future. It provides a single interface to all robots and devices and is can be customized for operations at your site.
- `samtool(1M)` - This X Window-based GUI consists of three interfaces: `robottool`, `devicetool`, and `previewtool`. These tools will be phased out in a future release of ASM.

To use the tools, you need to be familiar with using the OpenWindows environment pointer and SELECT and MENU mouse buttons. For information about working in the OpenWindows environment, see the *OpenWindows 3.1 User's Guide.*

This chapter contains the following:

- Operator privilege levels
- Using `libmgr`
- Using `samtool`
- Using `robottool`
- Using `devicetool`
- Using `previewtool`

Note: In addition to these GUIs, you can manage ASM using the curses-based operator tool, `samu(1M)`. Refer to Chapter 11, “Operator Information”, for instructions on using `samu(1M)`.

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 mount requests and changing device states. You can set up an operator group and define permissible operator tasks in the `/etc/fs/samfs/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 `defaults.conf` using the `operator` keyword. Privileged tasks for the operator group are defined using the `oper_privileges` keyword. Labeling of media, performing audits, moving media in a library, and changing device states are all examples of operator tasks that can be defined. See `defaults.conf`(4) for a complete listing of operator-privileged tasks.

The Library Manager

The Library Manager (invoked with the `libmgr(1M)` command) is a graphical user interface for managing media libraries. Using `libmgr`, administrators and operators can check the status of robots and media, import and export media, and respond to media mount requests.

Before using `libmgr`, make sure the LSCgui and LSCjre packages are installed on your system. (LSCjre is required only if there is no jre present on the server.) To check for these packages, enter the following:

```
server# pkginfo LSCgui
server# pkginfo LSCjre
```

If these packages are not installed, see Chapter 2, “ASM Installation and Configuration” for package installation instructions.

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/fs/samfs/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

See the `SamGUI.rsc(4)` manual page for a complete listing of resource settings.

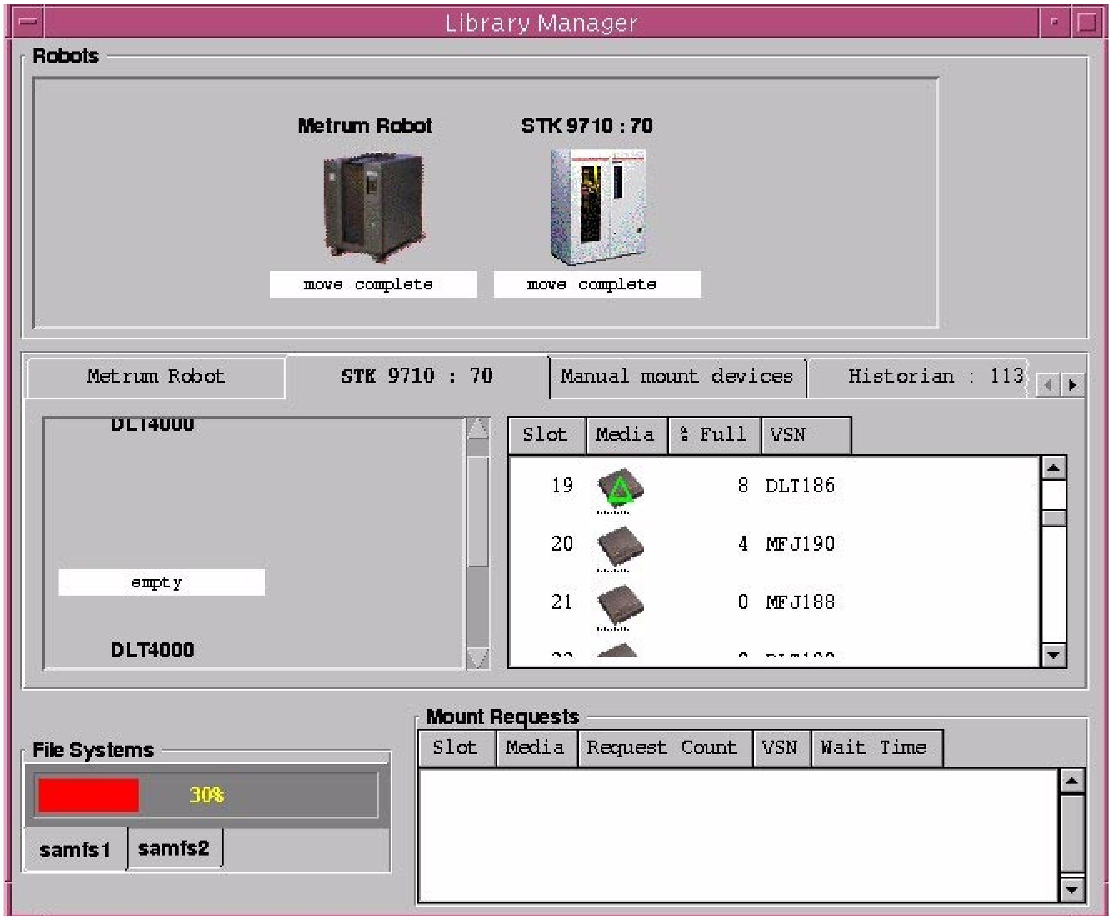
To reconfigure the `libmgr` displays, you must edit the SamGUI.rsc file, exit `libmgr`, and restart `libmgr`.

The Library Manager Display

An example of the Library Manager display is shown below. The display consists of objects that can be manipulated by a mouse. Most objects respond to the mouse as follows:

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, and the file systems and mount request panel on the bottom.



Library Manager
Robots
Metrum Robot
STK 9710 : 70
move complete
move complete
Metrum Robot
STK 9710 : 70
Manual mount devices
Historian : 113
DLT4000
empty
DLT4000
Slot
Media
% Full
VSN
19
8
DLT186
20
4
MFJ190
21
0
MFJ188
Mount Requests
Slot
Media
Request Count
VSN
Wait Time
File Systems
30%
samfs1
samfs2

Robot Operations

Place the pointer on the desired robot image, then:

Desired Action	Mouse Button	Menu Pick
Turn Robot 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

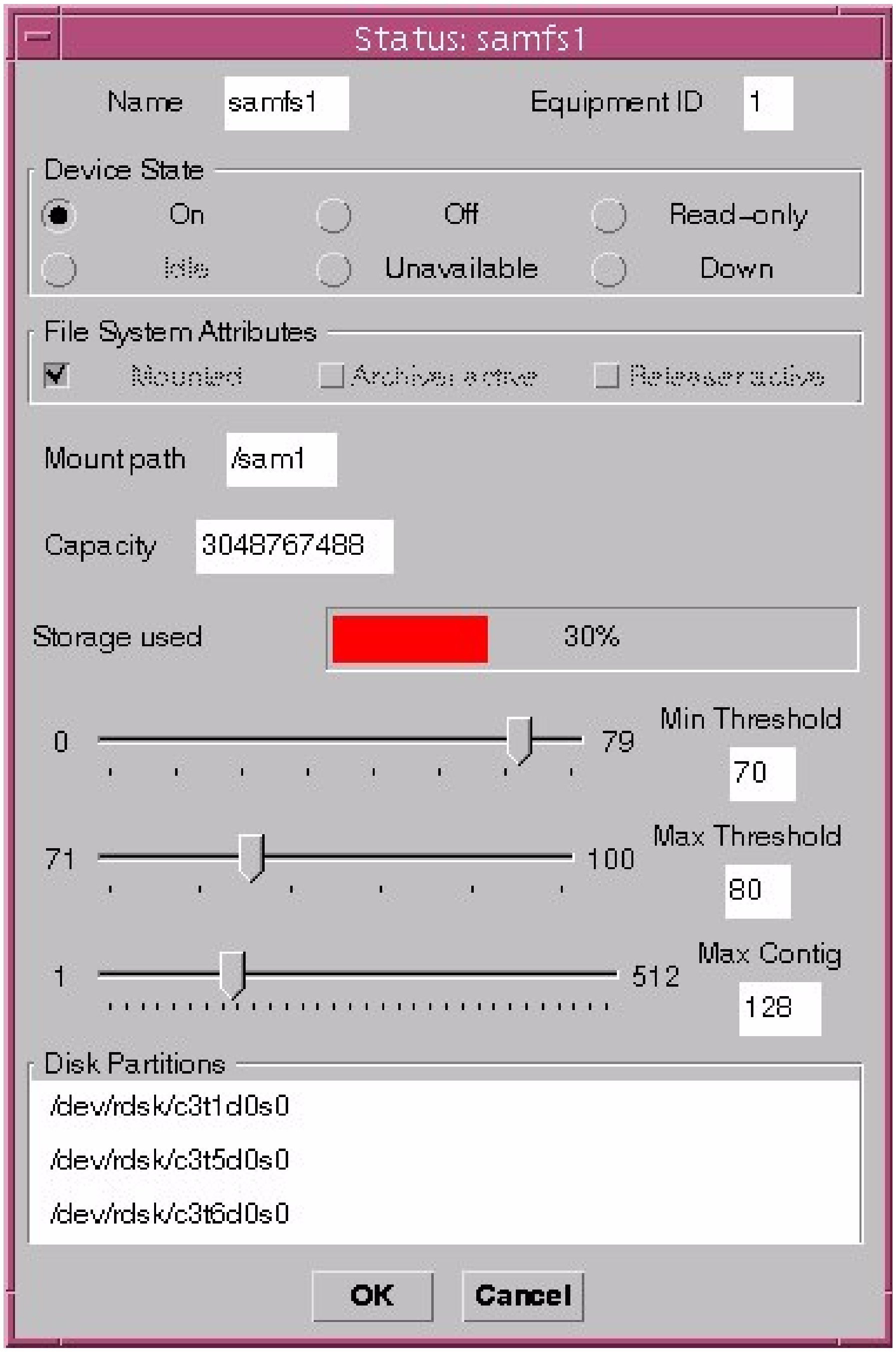
Media Drive Operations

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
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.

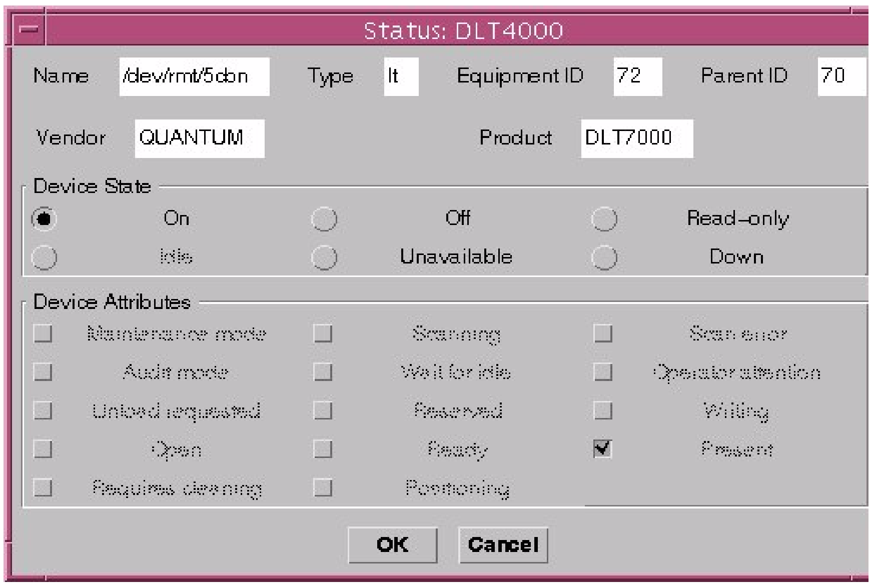


The file system device states and device attributes are described in the `libmgr(1M)` manual page under the “Icon Attributes” heading. Any changes to the attributes or states are enabled by selecting the button and clicking “OK”.

Note: Any changes to the minimum release threshold, maximum release threshold, and maximum contiguous block settings can only be set for the current ASM session. When the `sam-init` 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

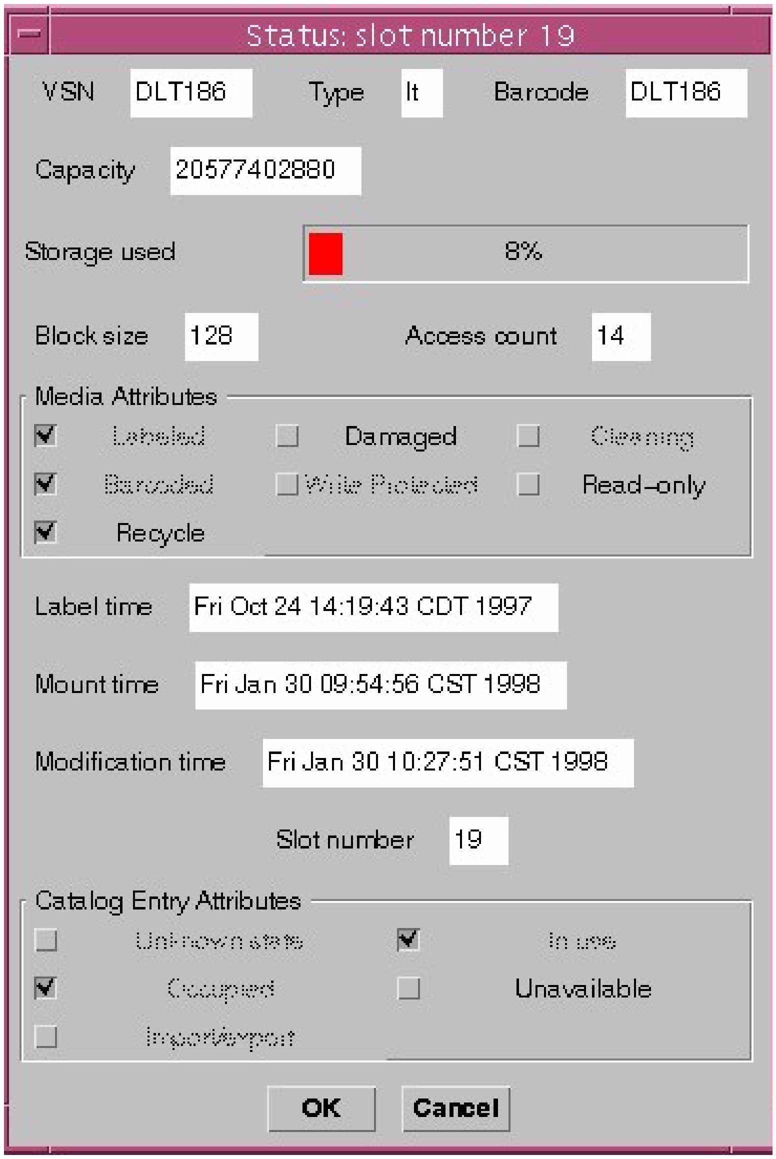
To view the media states and attributes and make changes, **double click** on the desired media drive image. The device state and attributes window for this device appears. An example follows:



The media drive device states and device attributes are described in the `libmgr(1M)` manual 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:



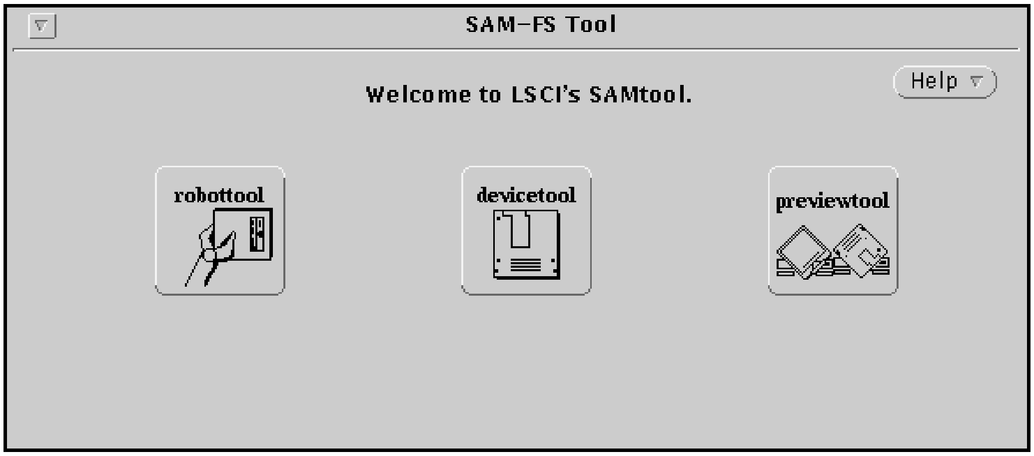
Using samtool

How to Start and Quit samtool

To start `samtoolmgr` enter the following command at the operating system prompt:

```
server# samtool&
```

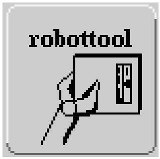
The system displays the `samtool` group, as follows:



To exit samtool, click MENU on the top bar of the window and select QUIT.

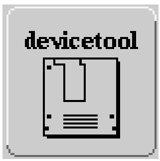
Selecting a Tool

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



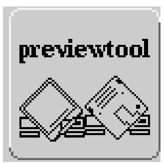
`Robottool`

Allows you to view and manual page information on robots associated with ASM.



`Devicetool`

Allows you to view and manage information on devices associated with ASM.



`Previewtool`

Allows you to view and manage pending mount requests associated with ASM.

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 the refresh checkbox to enable or disable automatic refresh.

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

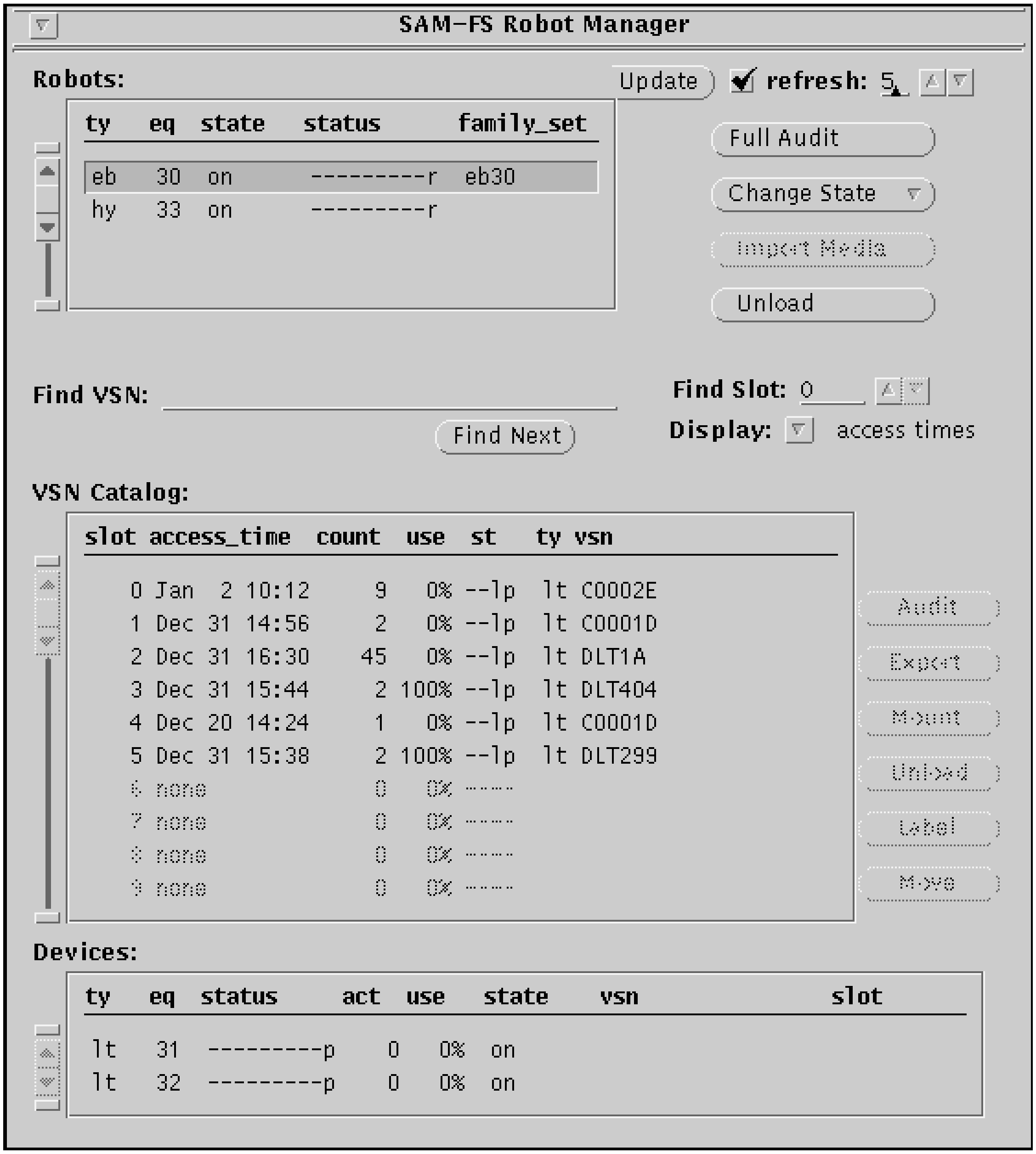
To display online help for `samtool`:

- Click SELECT on the Help button to display general help on `samtool` and its operation.
- Click MENU 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

Initially, `robottool` displays the first ASM robot. When you select a robot in the robot display, the system shows a VSN catalog and devices for the selected robot.

The following figure shows a sample `robottool` display:



The following table describes the areas of the `robottool` display:

Display Area	Description
Robots:	Lists all robots associated with ASM. The following information is displayed for each robot: **ty** Equipment type. **eq** Equipment ordinal. **state** State of equipment. See *Changing the State of a Robot* in this section. **status** Status of the robot. See *Viewing Status Information* in this section. **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	Lists the VSNs for the selected robot. The following information is displayed for each VSN: **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. **ty** Media type. **vsn** Volume serial name. You can use the VSN action buttons to Audit, Export, Mount, Unload, Label, and Move volumes.
Devices:	Displays information on devices for the selected robot. **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
`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	
`---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)	
`---------R`	device is ready and the media is read-only	
`---------r`	device is spun up and ready	the file system's disk space is being released
`---------p`	device is present	

Managing Robots

This section describes the actions you can perform on a selected robot:

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. Click SELECT 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:
   - Click SELECT on the Change State button to change the state to ON.
   - Click MENU on the Change State

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

Importing and Exporting Media

To import media into a robot:

1. Select the robot in the list of available robots.
2. Click the Import Media button.
3. Place the medium in the robot's mailbox.

The system instructs the robot to accept the piece of media placed in the robot's mailbox. When you have selected import media you can continue to place media in to the mailbox. If 30 seconds pass without inserting media the import operation is terminated.

To export media 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 medium in the robot's mailbox.

Note: You can import and export media 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 section 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

Tailoring the VSN Catalog Display

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:

- Click MENU on the Display button shown below and select barcodes:

The system displays barcodes rather than access times.

To display access times:

- Click MENU on the Display button and select access times.
  - The system displays access times rather than barcodes.

Selecting a VSN

To select a VSN:

- Click SELECT on the VSN you want to select.

Finding a VSN

To search for and select a volume by VSN:

- Type a VSN name or 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 results.

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. Click SELECT on the Audit button.

The system reads the VSN and updates the catalogue 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. Click SELECT 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. Click SELECT 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. Click SELECT on the Unload button.

The robot unloads the selected VSN from the robot's device and puts it back in the slot.

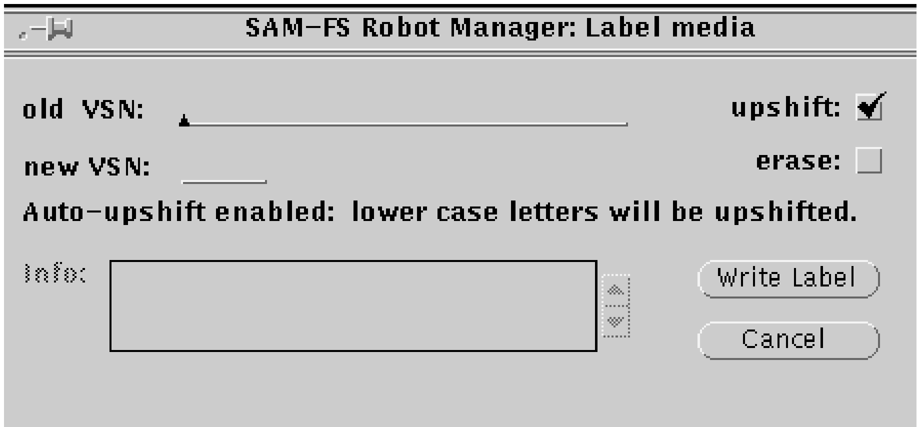
Labeling a Volume

Labeling a volume is required for ASM to distinguish one medium from another. A software label provides ASM with important information including the name of the VSN and the location where data begins to be written on the medium.

Note: 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. Click SELECT on the Label button. The following dialog is displayed:



3. Enter the following:

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.
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.

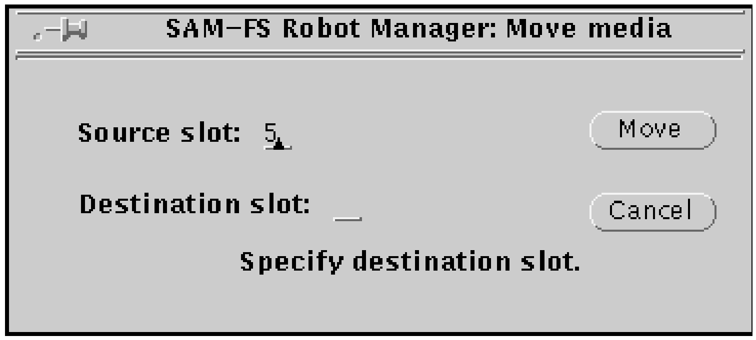
6. If an error is detected, an error checkbox and message appear in the Label media window above the info box. To acknowledge the error, click in the checkbox, and the error message is removed.

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

Moving Media

To move a volume to another slot:

1. Select the volume you want to move.
2. Click SELECT on the Move button. The following dialog is displayed:



3. Enter the following:

Source slot:	By default, the Source slot field contains the slot number of the volume you have selected. You can enter a new slot number by either: • Double-clicking on the slot number and type a new number. • Backspacing over the number to erase it and type a new number. The source slot you specify must contain a volume.
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 first column displays the status of the catalog entry. The following table describes the possible status flags:

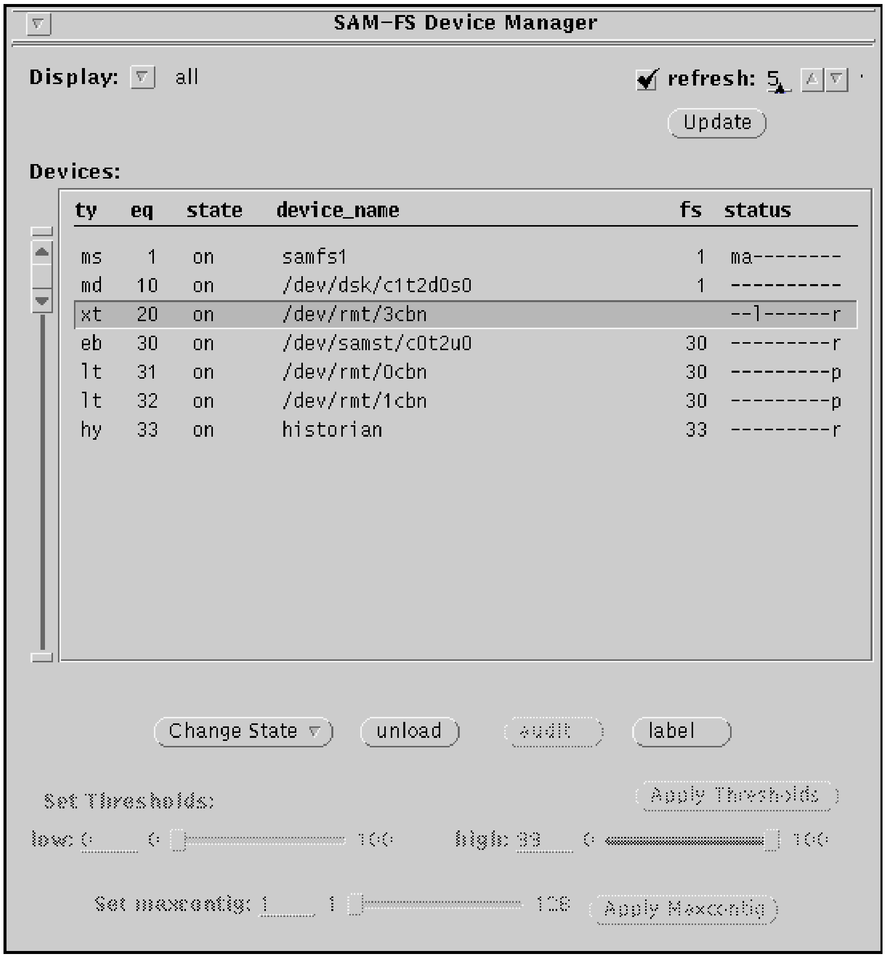
Status bit	Meaning
A---	Volume needs audit
R---	Volume is marked for recycling
W---	Volume is write protected
-E--	Bad media
-X--	This is an export slot
-r--	Volume is marked read-only
--u-	Slot is unavailable
--l-	Volume is labeled
---c	Cleaning
---p	Slot is occupied

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

The following figure shows a sample `devicetool display:`

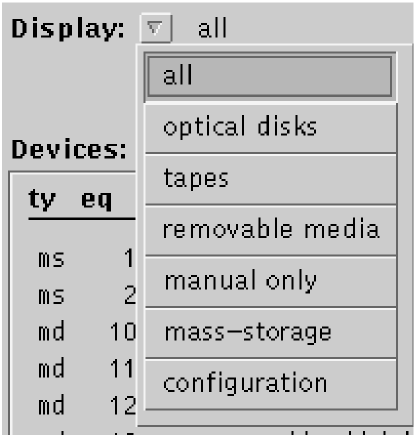

SAM-FS Device Manager
Display: all
refresh: 5
Update
Devices:
ty eq state device_name fs status
ms 1 on samfs1 1 ma--------
md 10 on /dev/dsk/c1t2d0s0 1 ----------
xt 20 on /dev/rmt/3cbn --l------r
eb 30 on /dev/samst/c0t2u0 30 ---------r
lt 31 on /dev/rmt/0cbn 30 ---------p
lt 32 on /dev/rmt/1cbn 30 ---------p
hy 33 on historian 33 ---------r
Change State
unload
audit
label
Set Thresholds:
Apply Thresholds
low: 0 0 100
high: 33 0 100
Set maxcontig: 1 1 128
Apply Maxcontig

Changing the Display Format

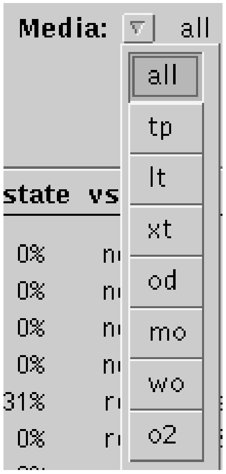
`devicetool` displays individual devices associated with ASM. The devices are displayed in a scrollable list in the center of the screen. By default, all devices are displayed.

To change the format of the display:

1. Click MENU on the Display button. The 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, click MENU on the Media button. The 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.
`maxc`	Maximum contiguous size of read/write operations in this file system (in units of 16KBytes).
`low/high`	Low and High disk usage thresholds percentage.
`ord`	Ordinal number of the disk device within the storage family set.
`state`	Current operating state of the device. Valid device states are: State Description **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.
`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.

Controlling Devices

This section 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:
   – Click SELECT on the Change State button to change the state to ON.
   – Click MENU 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. Click SELECT 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. Click SELECT on the Audit button.

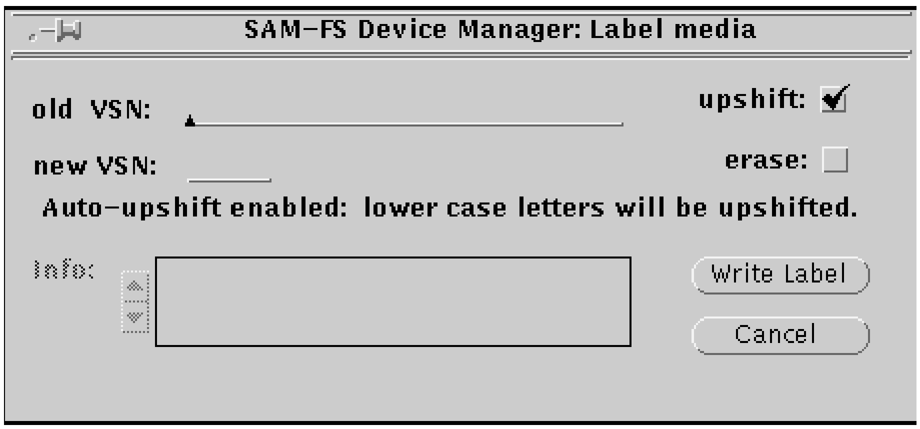
The system reads the volume in the device and updates the catalog entry.

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. Click SELECT on the Label button. The following dialog is displayed:



3. Enter the following:

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.

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 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.

Note: 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. Do the following:

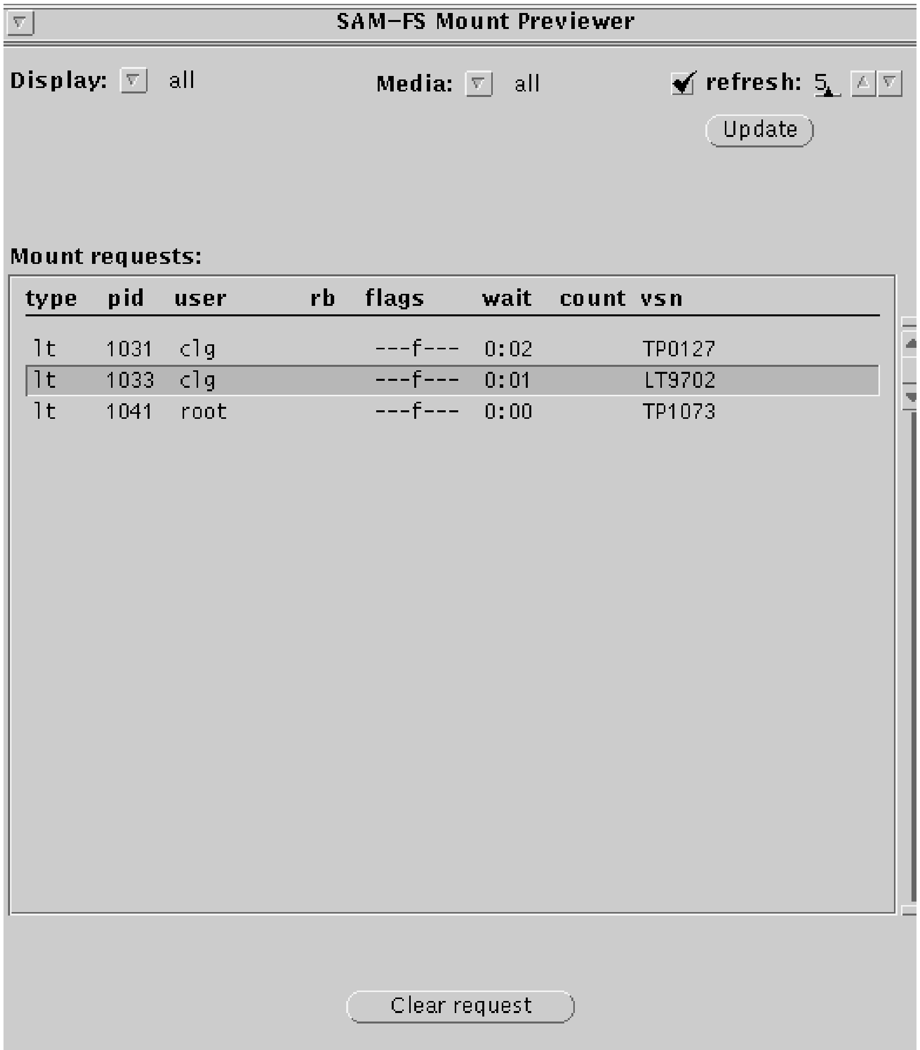
   – Set the Low threshold by either typing a number that specifies the percentage of use for the low threshold, or use the slider bar to increase or decrease the number.

   – Set the High threshold by either typing a number that specifies the percentage of use for the high threshold, or 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.

Using previewtool

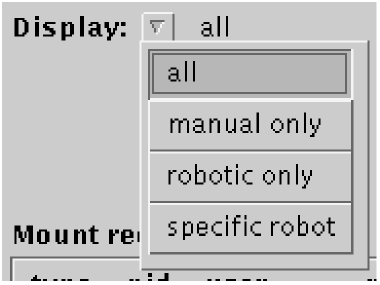
`previewtool` allows you to view and manual page pending mount requests. Initially, the display shows all pending mount requests in the mount request window. The information is displayed as a scrolling list. The window can also be resized to show from one to 18 mount requests by grabbing the window corner and stretching or contracting. The following figure shows a sample `previewtool` display:



Changing the Display Format

To change the display format:

1. Click MENU on the Display button. The following options are 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:

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 *Viewing the Flags Field* below.
`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.

Viewing the Flags Field

`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-`	flipside 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. Click SELECT on the Clear request button.

Chapter 11. Operator Information

Introduction

This chapter provides instructions for operating ASM via its operator utility (`samu`). This chapter includes the following:

- Using the ASM operator utility
- Operator commands
- Operator displays

Using the ASM Operator Utility

The ASM operator utility requires a display terminal that displays a minimum of 24 lines by 80 characters wide. The utility provides the following:

- Displays that allow you to monitor ASM devices and other file system activity.
- A set of commands that allow you to select displays and set display options, control access 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 configuration of your ASM.

The following sections describe how to start and stop the ASM operator utility, interact with the utility, access the help windows, and view operator displays.

Starting the ASM Operator Utility

To start the ASM operator utility enter the `samu` command from the UNIX command line as follows:

```
server# samu
```

The system starts the ASM operator utility 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 can accept options on its command line including selection of its initial display. See the `samu(1M)` manual page for more information.

Note: `samu`, like the `vi` editor, is curses based. You must have your terminal type defined correctly before starting `samu`.

Stopping the ASM Operator Utility

To exit the ASM operator utility session, enter one of the following:

- Press the **`q`** key
- Enter **`:quit`** or **`:q`**

The system exits the ASM operator utility and returns you to the command shell.

Interacting with the ASM Operator Utility

Interacting with the ASM operator utility is similar to interacting with the UNIX `vi(1)` editor with respect to paging forward/backward, entering commands, refreshing the display, and quitting the utility.

While viewing an operator display, the keys described below can be used to control the display. The exact function of these keys is dependent upon the display being viewed at the time. The descriptions shown in the following table are the typical responses.

Key	Description
`q`	Quit the ASM operator utility.
`:`	Enter a command. Command separators are space and comma.
`space`	Refresh the display.
`Ctrl-f`	Page forward. For equipment-oriented displays, advance to the next equipment.
`Ctrl-b`	Page backward. For equipment-oriented displays, return to the previous equipment.
`Ctrl-d`	Half page forward. For displays that do not support half-paging, advancement is the same as **`Ctrl-f`**.
`Ctrl-u`	Half page backward. For displays that do not support half-paging, advancement is the same as **`Ctrl-b`**.
`Ctrl-k`	Advance the display format. In the v (VSN catalog) display, advance the sort key. Sort selections for the v display are: slot, count, usage, VSN, access time, barcode, and label time.
`Ctrl-l`	Refresh the display (clear).
`Ctrl-r`	Toggle the display refresh function on or off (default is refresh on).

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

Many `samu` commands reference a specific device. For example: the command **`:off`** ***`eq`***. The ASM device *eq* is identified by its equipment ordinal (for example, 10). The equipment ordinal is defined in the `mcf`.

At certain times `samu` prompts for a device to be entered. For example, 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 as described above, or enter a carriage return to select the previous device used.

Getting On-line Help

When you start the ASM operator utility, the system automatically displays the first help screen. To move forward or backward from one screen to the next:

- 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

Operator displays are viewed by simply pressing the key corresponding to each display. The lowercase keys `a` through `v` display operational information. For descriptions of these keys, see, "Operator Display Keys" following.

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.

Operator Display Keys

Key	Description
a	Displays archiver status information.
c	Displays ASM 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.
p	Displays removable media mount requests. • Enter only **p** to display mount requests for all removable devices currently selected. • Enter **:p** *dt* 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** *dt* 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 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---------`		the file system is currently mounted
`M---------`	Maintenance mode	
`-E--------`	device received an unrecoverable error in scanning	
`-a--------`	device is in audit mode	the file system is being archived
`--l-------`	media has a label	
`---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-`	all storage slots occupied (robot status only)	
`---------R`	device is ready and the media is read-only	
`---------r`	device is spun up and ready	the file system's disk space is being released
`---------p`	device is present	
`---------N`	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 maybe superseded by the states **`ready/notrdy`**.
`ro`	The device is available for read-only access. Like on, this state maybe superseded for certain displays by **`ready/notrdy`**.
`off`	The device is not available for access.
`down`	The device is available for maintenance access only.
`idle`	The device is not available for new connections. Operations in progress continue until completion.
`ready`	The device is on and the disk or tape loaded in the transport is available for access.
`notrdy`	The device is on but no disk or tape is present in the transport.
`unavail`	The device is unavailable for access and may not be used for automatic ASM operations. The operator or system administrator may continue to use the `load` and `unload` commands for placing and removing media from the device while it is in the `unavail` state.

Operator Commands

This section describes the following operator commands:

- Display control commands
- Device commands
- Robot commands
- Miscellaneous commands

Note: Operator commands must be entered from the ASM operator utility. These commands are invalid if you enter them from the UNIX command line.

All `samu` operator 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

: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. When the *`filesystem`* argument is used the archiver status display shows the number of regular files, off-line files, archived files, archive copies and directories, plus the file systems, mount points, inode activity, and interval.

For the following commands, the valid medium types are listed in the `media(4)` manual page and in the `h` (help) display:

Code	Description
all	All removable device requests
at	Sony AIT tape
dt	4mm digital tape (DAT)
d2	Ampex DST310 (D2) tape
d3	STK SD3 tape
fd	Fujitsu M8100
ib	IBM 3590 tape
i7	IBM 3570 tape
lt	Digital linear tape (DLT)
mo	5 ¼” erasable optical disk
od	All optical disk requests
o2	12” WORM optical disk
se	STK 9490 tape
sg	STK 9840 tape
so	Sony DTF tape
st	STK 3480 tape
tp	All tape requests
vt	Metrum VHS tapes
wo	5 ¼” WORM optical disk
xt	Exabyte 8mm tape requests

:p `media`

The `p` command selects the media type for the mount requests display. *media* is the media type. 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. *media* is the media type. 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. *media* is the device type. 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 `:v historian`. If *`eq`* is not specified, the utility displays the catalog for the currently selected library.

Device Commands

:on `eq`

The `on` command logically turns on the specified device *eq*, where *eq* is the equipment ordinal.

:idle `eq`

The `idle` command restricts access to the specified device *eq* by preventing new connections to the device. Existing operations continue until completion. *eq* is the equipment ordinal.

:off `eq`

The `off` command logically turns off the specified device. *eq* is the equipment ordinal.

:down `eq`

The `down` command terminates operation on the specified device *eq*, where *eq* is the equipment ordinal.

:maxcontig `eq value`

The `maxcontig` command specifies the maximum number of bytes that can be read ahead or written behind by the file system. *eq* is the equipment ordinal for the file system. *value* is in units of 16k blocks and must be an integer from 1 to 512. The default *value* 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:

```
maxcontig 3 16
```

`Maxcontig` is also configurable as the `readahead` parameter in the ASM file system configuration file, `samfs.cmd`. See the `samfs.cmd(4)` manual page for more information.

:partial `eq partial`

The `partial` command sets the number of kbytes to leave online after release of the file. *eq* is the equipment ordinal for the file system and *partial* is the number of kilobytes to leave online. The default *partial* value is 16k.

: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 to ASM.

: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 7, “The Releaser”, Chapter 10, “Window-based Administration Tools”, and Chapter 14, “Advanced Topics”.

:devlog `eq [event . . .]`

The `devlog` command sets device logging options. *`eq`* is the equipment number of the device from the `mcf` 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`

Robot Commands

: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`

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. *`eq`* is the equipment ordinal and *`slot`* is the decimal slot number.

:import `eq`

The `import` command causes the specified robotic device to allow you to add a cartridge. *`eq`* is the equipment ordinal.

:mtslot `eq slot`

The `mtslot` (mount by slot) command causes the specified robotic device to load the volume into a drive. *`eq`* is the equipment ordinal or device name and *slot* is the decimal slot number containing the volume you want to load.

:mtvsn `eq vsn`

The `mtvsn` (mount by VSN) command causes the specified robotic device to mount a labeled volume in to a drive. *`eq`* is the equipment ordinal or device name. *`vsn`* is the volume to mount.

Archiver Commands

: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. See the `archiver.cmd(4)` manual page for a complete listing of all trace options.

Miscellaneous 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 toggles display refreshing on and off.

:snap `filename`

The `snap` command sends a snapshot of a display window to `filename`, the name of the file specified to receive the display information. The default file name is `snapshots`.

:clear `vsn [slot]`

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. *`slot`* is optional. If *`slot`* is specified, *`slot`* is the decimal ordinal of the VSN in the removable media display.

:quit

:q

The `quit` command (`q` for short) exits the `samu` operator utility.

:! `shell-command`

The `! command` allows you to run a shell command without leaving samu.

Operator Displays

This section provides descriptions and examples of the operator displays. Where 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 continues on an additional display page.

xb54	54	exb8505	pt03	0	yes	2	0	on	
lt55	55	dlt2000	pt02	1	yes	4	0	on	ml65
hp56	56	hpc1716	pt01	1	yes	3	0	on	hp70
hp57	57	hpc1716	pt01	1	yes	4	0	on	hp70
more									

Screen Snapshots

To provide a mechanism to aid the customer in reporting problems, all `samu` displays can have a snapshot of the screen taken. Screen snapshots are taken by entering the **:snap** command. Each new snapshot is appended to the snapshots file. The default file is `snapshots` in the current working directory. Optionally a different file can be specified on the snap command (for example, **:snap filename**). The file can be printed, re-examined using `vi`, or faxed to StorageTek support personnel.

(a) - Archiver Display

The archiver display shows status of the archiver on a per file system basis. To view the ASM 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    3.3.1-
L  Fri Jun 04 14:36:08
License: License never expires.

samfs1 mounted at /sam1
regular files      1,358    1.3G
offline files        681  650.9M
archdone files     1,357    1.3G
copy1              1,354    1.3G
copy2              1,354    1.3G
copy3                  0
copy4                  0
directories            8   76.3k
Sleeping until Fri Jun 04 14:37:30 1999
```

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 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
`directories`	Number of directories and total size
`sleeping until`	Indicates when archiver runs again

(c) - Configuration Display

The configuration display shows the connectivity of your ASM configuration. To view the ASM configuration display, press the **c** key.

Sample Display

```
Device configuration:                    samu 3.3.1-
L    Fri Jun 04 14:36:27
License: License never expires.
```

Device configuration: License: License never expires.		samu 3.3.1-L	Fri Jun 04 14:36:27
ty	eq state	device_name	fs family_set
ms	1 on	samfs1	1 samfs1
md	2 on	/dev/dsk/c0t0d0s0	1 samfs1
md	3 on	/dev/dsk/c0t0d0s1	1 samfs1
md	4 on	/dev/dsk/c0t0d0s6	1 samfs1
md	5 on	/dev/dsk/c0t2d0s0	1 samfs1
md	6 on	/dev/dsk/c0t2d0s1	1 samfs1
md	7 on	/dev/dsk/c0t2d0s6	1 samfs1
sk	50 on	/etc/fs/samfs/stk50	50 sk50
sg	51 on	/dev/rmt/1cbn	50 sk50
sg	52 on	/dev/rmt/2cbn	50 sk50
sk	60 on	/etc/fs/samfs/stk60	60 stkdlt
lt	61 on	/dev/rmt/0cbn	60 stkdlt
tp	62 off	/dev/rmt/3cbn	60 stkdlt
hy	63 on	historian	63

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: State Description **`on`** The device is available for access. **`ro`** The device is available for read-only access. **`off`** The device is not available for access. **`down`** The device is available only for maintenance access. **`idle`** The device is not available for new connections. Operations in progress continue until completion.
`device_name`	Path to the device (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.

(l) - License Display

The license display shows the licenses and expiration dates for each LSC product. To view the ASM configuration display, press the **l** key.

Sample Display

The following figure shows a sample display of license information for an ASM storage system. The license information is derived from the license keys in `/etc/fs/samfs/LICENSE.3.3`. The following information is displayed for the system:

- Expiration information
- Host ID
- LSC products and features enabled
- Equipment/media combinations

```
License Information                              samu        Fri Jun 04
License: License never expires.                  3.3.1-L     14:34:35


hostid = 8073adf9

License never expires

Robot type STK ACSLS Library is present
and licensed

   20 lt slots present and licensed

   0 tp slots not licensed

Robot type STK ACSLS Library is not
licensed
```

(m) - 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 figure shows a sample display of the status for two file systems 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      3.3.1    Fri June 04
                                                               -L       14:34:25

License: License never expires.


ty       eq    status     use      ord   capacity      free   mcntg    part   high    low
                          state

ms       1     m------    70% on         965.688M  288.906M       8       8    80%    70%
               --r

    md   2     -------    87% on     0    32.141M     4.062M
               ---

    md   3     -------    88% on     1    32.141M     3.828M
               ---

    md   4     -------    67% on     2   418.562M   136.453M
               ---

    md   5     -------    87% on     3    32.141M     4.062M
               ---

    md   6     -------    88% on     4    32.141M     3.953M
               ---

    md   7     -------    67% on     5   418.562M   136.547M
               ---
```

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.
`high`	High disk usage threshold percentage.
`low`	Low disk usage threshold percentage.

(n) - 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 pathname 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 **lt**.

Removable media I/O activity: all License: License never expires.	samu	3.3.1-L count: 4	Fri Jun 04	14:37:0 1

ty	eq	pid	file_size	staged_in	fseq	ino	position	offset	
sg	51	26999	1.375M	0	1	64	0x12181	0xb8180 0	-p
sg	52	0	0	0	0	0	0	0	
lt	61	0	0	0	0	0	0	0	
tp	62	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.
`offset`	Offset of the archive file on the specific medium.

(o) - Optical Disk Status Display

The optical disk status display shows the status of all optical disk drives configured on the ASM. To view the optical disk status display, press the **o** key.

Sample Display

```
Optical drive status:      samu      Fri Feb   11:20:3
                                     9         3

     ty        eq    status       act       use    state      vsn

     mo        35    --l---         1       29%    ready    oper2
                       wo-r
```

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: State Description **`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.

(p) - Removable Media Mount Requests Display

The removable media mount requests display lists information on pending mount requests for removable media. You can select either a specific type of medium, 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 mount requests for the currently selected removable media, press the **p** key.

To view mount requests for all removable media, enter **:p all**.

To view mount 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 **control-k** key.

Sample Displays

```
Removable media mount requests: all both    Fri       Feb 9    11:21:
samu                                                           42

                                                count: 1

slot     type     pid     user     rb     flags     wait    count     vsn

  0       lt      473     root     40     Wb-f--    0:00              TAPE0
```

```
Removable media load requests all prioritymu          3.3.1-w25 Mon Apr 26
License: License never expires.                       21:44:27
                                                         count: 3

   slot      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
`slot`	Slot 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 mount.
`priority`	Priority of the request.
`rb`	Equipment ordinal of the robot in which the requested VSN resides.
`flags`	Flags for the device: `W------`write access requested `-b-----`entry is busy `--C----`clear VSN requested `---f---`file system requested `-----S-`flipside already mounted `------s`stage request flag
`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.

(r) - Removable Media Status Display

The removable media status display allows you to monitor the activity on removable medium 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 the currently selected removable media devices, press the **r** key. To view the status for all removable media devices, enter **:r all**. To view the status for a specific device, enter **:r** *`dt,`* where *`dt`* is the device.

Sample Display

```
Removable media status: all                 samu 3.3.1-L Fri Jun 04
                                            14:37:47

License: License never expires.


ty       eq       status             act      use      state     vsn

sg       51       --l----oPr         3        47%      ready     004977

                  Position with FSR           395

sg       52       ---------p         0        0%       notrdy

         empty

lt       61       ---------p         0        0%       notrdy

         empty

tp       62       ----------         0        0%       off

         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: State Description **`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, or nolabel if the volume is not labeled. This field is blank when no volume is present in the transport, or the device is off.

(s) - Device Status Summary Display

The device status summary display shows the status for all devices configured in the ASM. To view the device status summary display, press the **s** key.

Sample Display

```
Device                              samu 3.3.1-L Fri Jun 04 14:38:05
status

License: License never expires.


ty           eq state     device_name         fs status    pos

ms           1 on         samfs1              1 m--------


md           2 on         /dev/dsk/c0t0d0s0   1 ---------


md           3 on         /dev/dsk/c0t0d0s1   1 ---------


md           4 on         /dev/dsk/c0t0d0s6   1 ---------


md           5 on         /dev/dsk/c0t2d0s0   1 ---------


md           6 on         /dev/dsk/c0t2d0s1   1 ---------


md           7 on         /dev/dsk/c0t2d0s6   1 ---------


sk           50 on        /etc/fs/samfs/stk   50 m-------

             stk_mount(24) 0x201, volser 004972

sg           51 on        /dev/rmt/1cbn       50r--l----

             Staged 0x38800 of 0x703d1 inode 272

more
```

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 type ms, file system name).
`fs`	Equipment ordinal of the family set to which the device belongs.
`status`	Device status. For a description of status codes, see *Status Codes* in this chapter.
`pos`	Device position.

(t) - Tape Status Display

The tape status display shows the status of all tape drives configured in the ASM. To view the tape status display, press the **t** key.

Sample Display

```
Tape drive status                               samu 3.3.1-L Fri Jun 04 14:38:15

License: License never expires.


ty      eq      status              act     use     state     vsn

sg      51      --l----o-r          2       47%     ready     004977

                Staged 0x6d200 of 0x124133 inode 296

sg      52      s----R-o-r          1       37%     ready     nolabel

                Process labels: Rewinding

lt      61      ---------p          0       0%      notrdy

                empty

tp      62      ----------          0       0%      off

                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: State Description **`ready`** The device is on and the tape loaded in the transport is 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, or `nolabel` if the tape is not labeled.

(u) – Display All Files in Staging Queue

The samu “`u`” display lists all files in the staging queue. To select this display, type “`u`”. Press the control-k key to list the file pathname on the second line of each entry.

Following is a screen snap of the `samu` "`u`" display.

Sample Display

Stage queue for media type: all						samu 3.3.1-L Fri Dec 11 16:46:58		
License will expire Wed Feb 3 10:10:13 1999				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
`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.

(v) - 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 **control-k** key combination changes the sorting key for this display.

Sample Display

```
Robot VSN catalog by slot          : eq: 50samu 3.3.1-L Fri Jun 04 14:39:01
License: License never expires.                         count 100
  slot    access time       count     use  flags          ty vsn
    0
    2     1999/06/04 14:38  59        37%  -il---b-----   sg 004972
    3     1999/06/04 14:10  63        47%  -il---b-----   sg 004977
    4     1969/12/31 18:00  0          0%  -il-o-b-----   sg 004973
    5     1999/05/27 06:59  18       100%  -il-o-b-----   sg 004974
    6     1999/05/27 06:52  1          0%  -ilEo-b-----   sg 004975   MEDIA
                                                                      ERROR
    7     1999/05/27 13:41  35        31%  -il-o-b-----   sg 004978
    8
   99
```

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: `A-----------`Slot needs audit. `-i----------`Slot in use. `--l---------`Labeled. `---E--------`Bad media. `----o-------`Slot occupied. `-----C------`Volume is a cleaning tape. `-----p------`Priority VSN. `------b-----`Bar code detected. `-------W----`Write protect. `--------R---`Read only. `---------c--`Recycle. `----------U-`Slot unavailable.
`ty`	Device type.
`vsn`	Volume serial name of the volume.

Chapter 12. Application Programmer Interface (API)

Introduction

The ASM application programmer interface (API) provides a means for operations on an ASM file to be requested from within an application program and, optionally, from another machine in the network.

When a request is made, the process or program making the request is the *client* process or program, running on the *client* machine. The requests are received and processed by the *server*, running on the *server*, or *host*, machine. For the ASM API, the server machine is always the machine where ASM is running.

In the simplest case, the client and server machines are the same, and no network communication is necessary. In other cases, the application programmer will want to allow for the client program to run on a machine where ASM is not running; in this case, networked library calls must be used.

Two API libraries are available with ASM: `libsam and libsamrpc.` The library calls in `libsam` do not perform network communication; they make local requests only. In this case, each library call makes a system call, and the "server" is the local operating system.

The library calls in `libsamrpc` use Remote Procedure Calls (RPC) to communicate with a special server process. Because of the RPC mechanism, the client and server can exist on the same machine or on different machines in the network. The server process always runs on the machine where ASM is running.

Both `libsam` and `libsamrpc` are released in shared object `(.so)` and archive `(.a)` format for Solaris platforms. `libsam.so` and `libsam.a` are in `/opt/LSCsamfs/lib.` `libsamrpc.so and libsamrpc.a` are installed in `/opt/LSCsamfs/client/lib,` with symbolic links to them in `/opt/LSCsamfs/lib.`

Library Calls

These are the API library calls. Note that sam_opencat, sam_getcatalog, and sam_closecat are not available using RPC:

Library Call	Description
sam_initrpc	Initializes the RPC connection (libsamrpc only).
sam_closerpc	Closes down the RPC connection (libsamrpc only).
sam_archive	Sets archive attributes on an ASM file.
sam_release	Releases and sets release attributes on an ASM file.
sam_stage	Stages and sets stage attributes on an ASM file.
sam_stat	Gets the status of an ASM file; follows symbolic links.
sam_lstat	Gets the status of an ASM file; does not follow symbolic links.
sam_cancelstage	Cancels a pending or in-progress stage on an ASM file.
sam_opencat	Opens the catalog for a media library.
sam_getcatalog	Obtains a range of catalog entries from the catalog for a media library.
sam_closecat	Closes the catalog for a media library.

For more details about each of the library calls, see the corresponding manual page. Library calls contained in libsam are found in section 3 of online manual pages. Library calls contained in libsamrpc are found in section 3x of online manual pages. For example, to view the manual page for the libsamrpc library call sam_stat online, perform the following command:

```
man -s3x sam_stat
```

Using libsamrpc

The source code for `libsamrpc` is included in the release for customers who wish to write and run application programs on platforms that do not run Solaris; in this case, the library will have to be ported to the client machine. The source code is located in `/opt/LSCsamfs/client/src,` with example application programs in `opt/LSCsamfs/client/examples.`

Specifying the Server Machine

A call to `sam_initrpc` is required before any other RPC client API calls can be executed successfully. Only one `sam_initrpc` call is required, followed by any number of other client API calls (other than `sam_closerpc).`

`sam_initrpc` takes one argument: a pointer to a character string, which specifies the name of the server machine. If this pointer is NULL, `sam_initrpc` then checks for an environment variable named `SAMHOST`. If this environment variable is set, that name is used for the server machine. If there is no `SAMHOST` environment variable, the default server name `samhost` is used.

In summary, the name of the server machine can be specified in any of three ways, which are checked by `sam_initrpc` in the following order:

- argument to `sam_initrpc`
- environment variable `SAMHOST`
- default server name, `samhost`

RPC Server Process

The RPC API server process receives and processes the requests from the client. This server process, `/etc/fs/samfs/rpc.sam,` must be run on the same machine as ASM. `rpc.sam` must be running for client requests to execute successfully.

`rpc.sam` is started automatically by `sam-init` if the appropriate entry is made in `defaults.conf`. (See the sub-section “Configuring the API”.) It can also be started manually; it should be run as superuser. `rpc.sam` takes no arguments.

`rpc.sam` services the requests it receives by making the appropriate system call on the ASM machine and then returning the output or result to the client.

Configuring the API

The following steps describe setting up the API server and clients. These steps assume that ASM is properly configured and running.

Step 1: Configure the API Server

For the server portion of the API to run successfully, the RPC program name and number pair must be known on the server machine, and must be the same as the pair used on the API client machines.

Make an entry for the RPC program name and number. The RPC program number is a number chosen by you. The RPC program name is `samfs.` The name and number pair must be the same on the server and all clients. The `/etc/nsswitch.conf` file determines where you should specify the RPC program name and number pair (see `nsswitch.conf(4)`).

In `/etc/rpc` (or the NIS database), add the following line:

```
samfs 150005
```

In `/etc/services` (or the NIS database), add the following line:

```
samfs 5012/tcp # SAM-FS API
```

The API server will be started automatically by `sam-init if the following entry is made in the defaults.conf` file. Note that changes to the `defaults.conf` file do not take effect until the next time `sam-init` is initialized.

```
samrpc = on
```

`rpc.sam` will not be automatically started if no entry for it appears in the `defaults.conf` file, or if the following entry appears in the file:

```
samrpc = off
```

See `defaults.conf(4)` for more information about the `defaults.conf` file.

Step 2: Configure the API Client Machines

Two configuration components must be present on the client machine for the RPC communication to be successful: the name of the server machine, and the RPC program name and number pair.

Make an entry for the RPC program name and number on all client machines, as you did on the API server machine above. Again, the RPC program name must be `samfs.` The RPC program number is a number chosen by you, but it must be the same on the server and client machines.

In `/etc/rpc` (or the NIS database), add the following line:

```
samfs 150005
```

The hostname of the server machine must be known on the client machine. For default cases, the hostname `samhost` must be listed as an alias for the ASM server machine. See `sam-initrpc(3X)` for more information.

Chapter 13. How to Update Tape Drives, Libraries, and Servers

Introduction

This chapter describes how to upgrade the hardware on your existing ASM storage server. The following topics are presented:

- Preparing for an upgrade
- How to add slots in a library
- How to upgrade or replace a media library
- How to upgrade DLT tape drives
- How to add disk cache to an ASM file system
- How to replace disks in an ASM file system
- How to upgrade a server

Preparing for Hardware Upgrades

Whether adding a new tape drive, adding a library, or installing a different drive into an existing media library, it is best to begin planning before the zero hour occurs. This section is intended to prepare you for the hardware upgrade of your ASM storage server.

The following are recommended actions prior to the upgrade:

- Determine if the hardware addition or change requires a new ASM license.
- Read the hardware manufacturer's installation instructions carefully. Also, you should read the section on adding hardware in the Solaris System Administration Guide.
- The system should be quiet with no users logged in.

- Ensure that all of the files that need to be archived have an archive copy. Enter the following for each ASM file system to see which files do not have an archive copy. (In this example, `/sam1` is the mount point.)
  ```
  # sfind /sam1 \!-archived \!-empty -type f -print
  ```
- The archiver must be in a wait mode (not running). You can idle the archiver by inserting a "wait" command into the `/etc/fs/samfs/archiver.cmd` file (see "Archiver Operations").
- The control structure information for all ASM file systems should be backed up prior to upgrading. See Chapter 9, "Dumping and Restoring ASM Control Structures".
- Check the equipment ordinals between your old and new master configuration files (see the `mcf(4)` manual page). If you did not define an entry for the `historian(7)`, ASM creates one by adding 1 to the highest equipment ordinal defined in the old mcf. Adding an equipment entry in the new mcf that has a conflicting ordinal with the old historian ordinal can produce undesirable configuration results.

How To Add Slots in a Library

The number of media slots managed by ASM is controlled by the license key supplied by StorageTek. To increase the number of ASM slots, follow these steps:

1. Obtain a new set of license keys through your StorageTek customer service engineer (CSE), or, if a CSE is not assigned to your account, through LSC, Inc. (See Chapter 2, "ASM Installation and Configuration" for details on obtaining a license.)
2. Replace the existing license keys with the new license keys. License keys start in column one `/etc/fs/samfs/LICENSE.3.3.` No other keywords, host ids, etc. may appear. The license becomes effective the next time `sam-init` is started.
3. Unload the library catalog using the `:unload` command in `samu(1M)`. This moves the robot's catalog entries into the historian catalog and preserves the catalog information for each piece of media. Catalog information can include the number of times it has been mounted, available space, etc. To move the catalog entries to the historian catalog, enter the following from `samu`:

   `:unload` ***`eq`***

   where *`eq`* is the equipment ordinal as defined in the `mcf` file. In `samu(1M)`, the robot's "`v`" display empties and the historian's "`v`" display fills up with the VSNs that used to be in the robot.

4. Bring down the ASM system. (See “How to Stop ASM” in Chapter 3, “ASM Basic Operations”.) Power down the server and library per the manufacturer’s suggested procedure.

5. Have the library hardware engineer add slots to the library. Power on the system using your normal startup procedure.

6. Start ASM. (See “How to Start ASM” in Chapter 3, “ASM Basic Operations”.) The new license information appears in the `samu` “l” display.

How To Upgrade or Replace a Media Library

Prior to disconnecting and installing a different tape or magneto-optical library, prepare for the upgrade as described in the “Preparing for Hardware Upgrades” section.

To replace or upgrade a media library, follow these steps.

1. Unload the library catalog using the `:unload` command in `samu(1M)`. This moves the robot's catalog entries into the historian catalog and preserve the catalog information for each piece of media. Catalog information can include the number of times it has been mounted, available space, etc. To move the catalog entries to the historian catalog, enter the following from `samu`:

   `:unload` ***`eq`***

   where *`eq`* is the equipment ordinal as defined in the `mcf` file. In `samu(1M)`, the robot’s “`v`” display empties and the historian's "`v`" display fills up with the VSNs that used to be in the robot.

2. Update the `/etc/fs/samfs/inquiry.conf` file, if needed. The new library should be identified in this file by the vendor, the library model, and by an ASM internal name. For example, the released `inquiry.conf` file has a line:

   ```
   "HP",   "C1710T",  "hpoplib"   # HP optical library
   ```

   This line indicates that if ASM detects a SCSI device made by vendor "HP" of model "C1710T", ASM drives it as an "hpoplib". The first two fields (vendor/product) are returned from the hardware devices; the last field "hpoplib" is a name ASM 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` as `/etc/vfstab.cur`. Edit the `/etc/vfstab` file changing any ASM mounts from "delay" to "no".

4. Save the `/etc/fs/samfs/archiver.cmd` as `archiver.cmd.cur`. Edit the `/etc/fs/samfs/archiver.cmd` file adding a “wait” command as the first line.

5. Power down the server and peripherals using the manufacturer’s suggested procedure. Disconnect the media library.

6. Attach the connecting cable to the new library and power on the peripherals and server using the suggested power on sequence.

7. Ensure that the server identifies the new robot. Enter “`probe-scsi-all`”. The new robot and its drives must be displayed prior to proceeding. If these devices are not identified, the robot 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 robot change, or the ordering or number of the drives in the robot changes, make sure you modify the `/etc/fs/samfs/mcf` to reflect the new configuration. This is similar to an initial installation described in Chapter 2, “ASM Installation and Configuration”.

   If adding new equipment, you may need to create new `/dev/samst` entries. Enter the following:

   ```
   # samdev
   ```

9. After the appropriate `mcf` changes and /dev/samst entries are complete, initialize ASM by either mounting an ASM file system or starting the `sam-init` process at the command line. When ASM initializes, it notices that the number of slots in the library has changed. The system runs a full audit on the media library to update the library's catalog. A full audit must be completed before archiving is resumed.

   If the audit completes without problems, you can replace `/etc/vfstab` and `/etc/fs/samfs/archiver.cmd` with the pre-upgrade versions. (Use the saved `/etc/vfstab.cur` and `/etc/fs/samfs/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 robot does not match the ordering in the `/etc/fs/samfs/mcf` file. Remember, drives have two attributes: SCSI target id, and position in the robot. The robot calls the drives by position number. When ASM wants to mount a medium in a drive, it must send the command to the robot to "mount medium from slot 123 into drive 3", for example.

   Drive 3 might be SCSI target 6 based on the third `mcf` entry. ASM knows it is drive 3 because it's the third drive entry in the `mcf` file. The robot knows that it is drive 3 because of the physical location it occupies in the robot.

   Once the robot has been requested to mount the medium into the drive, ASM starts using the SCSI bus to see if the drive goes ready. Here, ASM 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 medium.

   There is not a good way to determine this information. Most of the time, the manufacturer ships the robot 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) :mtslot` command to mount a medium, and then watch the `samu s` display to see which drive comes ready versus which drive has the "mounting" message.

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 a library or a stand-alone tape drive, such as moving from DLT 4000 drives to DLT 7000 drives. With ASM this is a matter of adding the new drive, rebooting the new configuration, and updating the `mcf(4)` file as necessary, prior to starting ASM.

The following restrictions and general information should be noted prior to upgrading drives:

- ASM does not handle mixed DLT tape drives within the same SCSI-attached robot. For example, ASM cannot differentiate between a DLT 4000 tape drive and a DLT 7000 tape drive in the same robot. 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 the 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. See Chapter 8, “The Recycler”, for information on recycling tapes.
- As tapes are labeled the density of the tape is acknowledged and recorded in the robot catalog.

To upgrade tape drives, follow these steps:

1. As a precaution, run `samfsdump(1M)` on the ASM file 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 a ASM internal name. For example, the released `st.conf` file has a 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. See Chapter 2, “ASM Installation and Configuration” for more information on updating `st.conf`.
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 "`probe-scsi-all`". The robot 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 robot change, or if the ordering or number of the drives in the robot changes, modify the `/etc/fs/samfs/mcf` file to reflect the new configuration. This is similar to an initial installation as described in Chapter 2, "ASM Installation and Configuration".

   New `/dev/samst` entries may need to be created for the new equipment. To create these entries, enter the following command:

   ```
   # samdev
   ```

7. Start ASM using your normal startup procedure and mount the ASM file systems. At this time, you can continue to use the existing ASM tapes.

How To Add Disk Cache to an ASM File System

At some point, you may want to add disk partitions or disk drives in order to increase the disk cache for an ASM file system. This is accomplished by updating the `mcf` file and using `samgrowfs(1M)` as described in this section. There is no need to re-make or restore the file system. Note that when adding a disk or partitions, the equipment ordinal of the historian is updated. (The equipment ordinal of the historian is automatically generated by ASM unless you specifically call it out. See the `historian(7)` manual page for more information.

To add disk partitions or drives to an existing ASM file system, follow these steps:

1. Unmount all ASM file systems.

2. Stop the `sam-init` process.

3. Edit the `/etc/fs/samfs/mcf` file. Add the new partitions *after* the existing partitions for the file system to which you want to add disk cache. Save the changes and quit the editor.

   If the equipment identifier name in `/etc/fs/samfs/mcf` has changed, the file systems can no longer be mounted. Instead, the following message is logged in `/var/adm/messages:`

   Warning: ASM superblock equipment identifier <*id*>s on eq <*eq*> does not match <*id*> in mcf

4. Run `samgrowfs(1M)` on the file system (named *samfs1* in this example) that is growing:

```
# samgrowfs samfs1
```

5. Start ASM and mount the file systems.

How To Replace Disks in an ASM File System

If an upgrade or replacement of disks or partitions for a cache with different disk partitions or an entire disk, such as a disk array, the ASM file system structures need to be backed up and recreated.

To replace disk partitions and recreate the ASM file system, follow these steps:

1. Create a control structure dump of each ASM file system to be upgraded, using a unique dump file name for each file system. See Chapter 9, "Dumping and Restoring ASM Control Structures". In this example, *dump_file* is the newly created dump structure.

   ```
   # samfsdump -T -f dump_file samfs-mount-point
   ```

   If any file warnings are issued, these files need to be backed up before unmounting the file systems.

2. Unmount all ASM file systems.

3. Stop the `sam-init` process.

4. Edit the `/etc/fs/samfs/mcf` file to include the new disk or partitions. Save the changes and quit the editor.

   If the equipment identifier name in `/etc/fs/samfs/mcf` has changed, the file systems can no longer be mounted. Instead, the following message is logged in `/var/adm/messages`.

   Warning: ASM superblock equipment identifier <*id*>s on eq <*eq*> does not match <*id*> in mcf

5. Make a new file system. Run `sammkfs(1M)` on the file system being created (named *samfs1* in this example):

   ```
   # sammkfs samfs1
   ```

6. Start ASM and mount the file systems.

7. Restore each ASM directory using the control structure dump file created in Step 1.

   ```
   # samfsrestore -T -f dump_file
   ```

How To Upgrade a Server

When it comes time to upgrade the server being used for ASM, you should take the following in to account:

- It is wise to move to the new server while the existing server is in operation. This allows time to install, configure, and test the new hardware platform with your applications.
- Moving to a new server is roughly equivalent to installing ASM for the first time: reinstall the ASM software; update the ASM configuration files (specifically the `mcf`, `/kernel/drv/st.conf`, and `/etc/fs/samfs/inquiry.conf`; copy your existing `archiver.cmd` and `defaults.conf` files to the new system; configure system logging; etc. Use the instructions in Chapter 2, "ASM Installation and Configuration" as a guide to re-installing ASM.
- The license key needs to be updated. License keys are tied to the CPU hostid. Replacing the CPU requires a new license.
- Before powering down the old server, use `samfsdump(1M)` and complete a control structure dump on each ASM file system. This dump file is used to recreate the file system on the new server. See Chapter 9, "Dumping and Restoring ASM Control Structures" for explicit instructions.

Chapter 14. Advanced Topics

Introduction

This chapter discusses advanced topics beyond the scope of basic system administration and usage. The following topics are presented.

- Device Logging
- Large file sizes
- Creating files using the `request` command
- Volume overflow
- Retrieving a file on another UNIX system
- Performance related topics
- Prioritize Preview Requests

Device Logging

The device logging facility provides device-specific error information used to analyze certain types of device problems. It can help to determine a failing sequence of events for a library, tape drive, or optical drive. Note that the device logging facility does not collect soft media errors (e.g., recoverable reads).

Device logging messages are written to individual log files. There is a log file for each library device, each tape and optical drive device, and one for the historian. The log files are located in `/var/adm/devlog`. The name of each individual log file is the same name as the equipment ordinal. For example, an ASM system with a single HP optical library with two optical drives would look like this:

/dev/samst/c1t5u0	40	hp	hp40	-	etc/fs/samfs/hp40_cat
/dev/samst/c1t4u0	41	mo	hp40	-	
/dev/samst/c1t6u0	42	mo	hp40	-	

In `/var/adm/devlog`:

4-jafs: pwd

/var/adm/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:

- Enter the `samset` command "`samset devlog eq event`".
- In the `defaults.conf` file located in `etc/fs/samfs/defaults.conf`, enter `devlog` *`eq`* `all`. *`eq`* is the equipment number of the device from the `mcf`, 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

See the `samset(1M)` manual page for more information.

When ASM starts up, it automatically sets the event type for each available device to "`default`". You can use the `samset` 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 logging (see the following command):

```
err, retry, syserr
```

If you suspect there is a problem with one of the devices used by ASM, 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 excess logging.

The device log information is also automatically collected when `info.sh` is invoked. This allows the ASM 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)` manual page. You can select the following events:

`all`	`none`	`default`
`detail`	`err`	`label`
`msg`	`mig`	`retry`
`stage`	`syserr`	`time`
`module`	`event`	`date`

Note that the device log messages are only available in English text.

Large File Sizes

With Solaris 2.5, ASM accepts very large files up to 2^{64} -1 bytes (2^{34} -1 Gbytes). When a file surpasses 4 Gbytes (4294918144 bytes to be precise), ASM begins processing the file using direct I/O. This large file size is possible with 64-bit addressing.

Warning: ASM data is written to archive using standard tar format. For disaster recovery purposes we provide the `gnutar` command for restoring data on any UNIX system. Note, however, that `gnutar` does not work with files larger than 8 Gbytes.

Once the 4 Gbyte threshold is reached, it is important to read and write on buffer boundaries of 512 bytes and buffer sizes of 512 byte multiples, as this is the size of a disk sector. This dramatically improves performance for reads and writes. It is strongly recommended that the I/O request size used is as large as is practical, such as 256 kbytes.

In Solaris 2.6, ASM supports 64-bit files for both paging I/O and direct I/O. The type of I/O is selectable using the Solaris `directio(3C)` function.

When using very large files, careful attention must be given to the size of disk cache available on the system. If you are trying to write a file that is larger than your disk cache, the program blocks since there is not enough disk space available to handle such requests. Also note that memory mapping very large files is not supported. For instance, `cp` does not work if the files are large.

Applications that use sparse files do not work with very large files. When initially writing a very large file, writes must occur sequentially from beginning to end. After the initial file is written it is possible to perform seeks and writes to rewrite portions of a file that have already been allocated.

Also, make sure that you use the LSC-provided `sls(1)` command to determine the length of a file or an archive copy of a file. In Solaris 2.5, the standard UNIX `ls` command cannot return accurate information on files above the UFS 2 Gbyte size limit; this has been fixed with Solaris 2.6. Also, use the LSC-provided `sfind`(1) command when using the `-size` search parameter.

How To Configure ASM for Large File Sizes

ASM is tuned to work with a mix of file sizes. You may increase the performance of large disk file transfers by enabling the following system settings.

Step 1: Set the Maximum Read/Write Parameter

The `maxphys` parameter in the `/etc/system` file controls the maximum number of bytes the device drive reads or writes at any one time. Set `maxphys` as follows:

```
set maxphys = 0x800000
```

Step 2: Set the SCSI Disk Maximum Transfer Parameter

On Solaris 2.6 systems, the sd driver enables large transfers when tagged queuing is enabled by looking for the `sd_max_xfer_size` definition in `/kernel/drv/sd.conf`. If it is not defined, it used the value defined in SD_MAX_XFER_SIZE, which is 1024*1024 bytes.

To enable large transfers, set sd_max_xfer_size equal to the maxphys setting above. The following is segment from an example /kernel/drv/sd.conf:

```
name="sd" class="scsi"
 sd_max_xfer_size=0x800000
 target=2 lun=0;

name="sd" class="scsi"
 sd_max_xfer_size=0x800000
 target=3 lun=0;
```

Step 3: Reboot the System

To activate the above system settings, you must reboot the system.

Step 4: Set the ASM Readahead/Writebehind Parameter

The maxcontig setting specifies the maximum number of bytes that can be read ahead or written behind for an ASM file system. This setting is enabled either as a file system mount option (see mount_samfs(1M)) or as a command from samu(1M). The setting is in units of 16k blocks and must be an integer from 1 to 512. The default setting is 8 (131072 bytes). Maxcontig is also configurable as the readahead parameter in the ASM file system configuration file, samfs.cmd. See the samfs.cmd(4) manual page for more information.

Increasing the size of maxcontig increases the performance of large file transfers, but only to a point. You should test the performance of the system after resetting the maxcontig parameter until you see no more improvement in transfer rates. The following is an example method of testing timings on disk writes:

```
# timex dd if=/dev/zero of=/dev/rdsk/c0t3d0s2 bs=4096k
count=512
```

Creating Files Using the Request Command

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 *request files* or *removable media files*. Request files look like normal ASM files in that they have permissions, username, groupname, and size. 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.) ASM has that information in the inodes file. Multiple request files can reside on the same media.

If the file resides on optical disk, seeks can be performed on the file to randomly update the file. If the file resides on tape, the file must be written and read sequentially. The media type and at least one VSN for the media must be specified. Multiple VSNs (up to 256) can be specified to handle volume overflow (see "Volume Overflow" in this chapter). ASM automatically mounts the requested VSN if the VSN resides in a robot defined in the master configuration file.

The VSNs used for the `request` command should not be the same media that ASM uses for automated archiving. ASM archiving appends to the end of the current data the next file to be archived and moves the EOF label beyond the data each time.

Any request file on removable media prevents that VSN from being recycled. The `recycler(1M)` expects that archived files are the only usage of the particular VSN that is assigned for ASM archiving. In addition, the `request` files are never archived; the removable media is the only copy of the file. The type of data being written to request files is unknown to ASM.

Request files are not supported over NFS.

See the `request(1)` manual page for examples that describe how to create files using request.

Volume Overflow

Volume overflow allows the system to span a single file over multiple volumes. It is useful for sites using very large files that exceed the capacity of their chosen media. Note that when using the volume overflow feature, it may be difficult to retrieve volume overflow data if you need to run disaster recovery.

Enable volume overflow by setting the `ovflmin` parameter in the `archiver.cmd` file. When a file size exceeds `ovflmin`, the archiver writes another portion of this file to another available medium of the same type, if necessary. The portion of the file written to each volume is called a *section*. See the information on controlling volume overflow in Chapter 6, "Archiver Operations", for instructions on setting the `ovflmin` parameter for volume overflow.

Volume overflow files can be created directly by using the `request(1)` command. Note that using the `request` command bypasses the normal functions of the archiver. When overflowing the file to separate volumes, you must separate VSNs with a slash. See the `request(1)` manual page for the complete syntax. The following is an example `request` command creating a removable media file on Ampex D2 tapes using three volumes:

```
server> request d2 TAPE01/TAPE02/TAPE03 large.file
```

The `sls`(1) command lists the archive copy showing each section of the file on each VSN. The following example shows the archiver logfile and `sls -D` command output for a large file named **`file50`** that spans multiple volumes.

The archive logfile shown 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. See the `archiver`(1M) manual page for a complete description of the archiver log entry.

Archive logfile entry for **`file50`**:

A 97/01/13 16:03:29 lt DLT000 big.1 7eed4.1 samfs1 13.7 477609472 big/file50
1 97/01/13 16:03:29 lt DLT001 big.1 7fb80.0 samfs1 13.7 516407296 big/file50
2 97/01/13 16:03:29 lt DLT005 big.1 7eb05.0 samfs1 13.7 505983404 big/file50

And the `sls -D` output:

```
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
```

Note that volume overflow files do not generate checksums. See the `ssum(1)` for more information on using checksums.

Although the number of VSNs for a removable (request) media file has increased from 8 to 256, archiver can handle only 16 VSNs.

Retrieving a File on Another UNIX System

Archive data files can be retrieved even when ASM is not available. In this example, we are running on another UNIX system and need to recover data files on a tape. The drive is named `/dev/rmt/xcbn,` where *x* is an integer referring to the tape drive on which the tape is mounted, and *y* = blocksize/512. In this example, the drive is number 3. To extract the tar files from the tape, enter the following:

```
server> mt -f /dev/rmt/3 rewind              # Rewind tape to BOT
server> mt -f /dev/rmt/3cbn fsf 1            # Position past ANSI label
server> gnutar xvbf y /dev/rmt/3cbn          # Extract files on 1st tar file
server> gnutar xvbf y /dev/rmt/3cbn          # Extract files on 2nd tar file
server> gnutar xvbf y /dev/rmt/3cbn          # Extract files on 3rd tar file
server> . . .                                # Continue until EOT
```

If you are using volume overflow files, these steps do not work.

Performance Related Topics

Direct I/O

The `directio -D` option sets the direct I/O attribute for a file and/or directory and is used with the `setfa` and `sam_setfa` commands.

If applied to a directory, the direct I/O attribute is inherited. After the `directio –D` option is set, the file uses direct I/O. Data is transferred directly between the user’s buffer and the disk.

Set this attribute only for large block aligned sequential I/O. The default I/O mode is buffered (uses the page cache). Any I/O attempted on these files is direct.

Mount Parameters

Two mount parameters control the flush behind rate for pages written sequentially and stage pages. The `flush_behind` and `stage_flush_behind` mount parameters are read from `/etc/fs/samfs/samfs.cmd`. For more information on these mount parameters, see manual page `samfs.cmd(4)`.

flush_behind

The `flush_behind` command sets the maximum flush behind value to `n` (`n` is in kilobyte units). Modified pages that are being written sequentially are written to disk asynchronously to help the Solaris VM layer keep pages clean. `n` is an integer from 16 to 819. The default is 64 (65536 bytes).

```
flush_behind = n
```

stage_flush_behind

The `stage_flush_behind` command sets the maximum stage flush behind value to `n` (`n` is in kilobyte units). Stage pages that are being staged are written to disk asynchronously to help the Solaris VM layer keep pages clean. `n` is an integer from 16 to 8192 The default is 64 (65536 bytes).

```
stage_flush_behind = n
```

Prioritize 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 order (FIFO).

The system administrator is allowed considerable flexibility in assigning different priority to preview requests. The system administrator can override the FIFO default by entering commands in the preview command file located in `/etc/fs/samfs/preview.cmd`) to control scheduling preview requests based on whether the request is for file staging or archiving, and to increase the priority for specific VSNs. Further, settings in the `preview.cmd` can also reprioritize preview requests for all or specific file systems based on the high water mark (HWM) or low water mark (LWM) settings.

The system administrator specifies priority values and factors in the `preview.cmd` configuration file (see the manual page for `preview.cmd(4)`). The commands are read by `sam-init` at startup and the specified values are stored in shared memory. The commands are listed one per line. Changes made to this file while `sam-init` is running do not take effect until `sam-init` is restarted. The equal sign (`=`) must be preceded and followed by a space. Comment lines begin with a hash sign (`#`) and extend through the end of the line.

There are two types of commands: global commands and file system-specific commands. Global commands apply to all file systems and appear before the first "`fs =` " line. File system-specific commands follow the global command and begin with "`fs =` ". File system commands apply until either the next "`fs =` " line or the end of file is encountered.

Note: When multiple commands impact a file system, the file system-specific commands override the global commands.

Preview Request Commands

The numeric priority of preview requests is determined by combining several static or dynamic factors. 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.

Global Commands

`vsn_priority = value`

This command is a static priority factor that 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 (e.g., `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 global command is a static priority factor whose overall effect is dynamic. The `age_priority` factor is multiplied by the number of seconds a request is a preview request and 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 actually become higher than a VSN that comes into the queue later with the priority flag already set.

Water Mark Preview Request Commands

The following water mark priority commands determine the water mark priority (`wm_priority)` of the preview requests. The water mark priorities are used only to calculate media requests for archiving; they are not used to calculate media requests for staging. Together, the four water mark settings create a dynamic priority factor comprised of the "percent-full" of a file system 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, file system-specific, or not set at all.

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 `wm_priority` setting, with or without the "`p`" option. The calculation depends on the water mark and the direction from which it is crossed (LWM, LHWM, HLWM, and HWM).

`lwm_priority = value`

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

`lhwm_priority = value`

> Indicates 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 file system is filling up. The default for `lhwm_priority` is `0.0`.

`hlwm_priority = value`

> Indicates 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 the releaser was not able to free enough disk space to leave the file system below LWM. The default for `hlwm_priority` is `0.0`.

`hwm_priority = value`

> This priority command indicates the value by which the `wm_priority` factor changes for archiving requests when the file system is above the HWM level. The default for hwm_priority is `0.0`.

Positive and Negative Factors

When the wm_priority contributes a positive factor (i.e., the value 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 negative (i.e., the value is 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. See the example in the sub-section "Enforcing Stage Requests Prioritized over Archival Requests".

Difference between LHW and HLWM File system States

The following example shows how to slightly increase the priority for archiving requests when the file system is at HLWM. This example allows the releaser to free enough disk space so that the file system gets below LWM.

```
lhwm_priority = -200.00
hlwm_priority = 100.00
```

Calculating Total Preview Request Priority

The total priority for a preview request is the sum of all priority factors and 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. See the following list of possible conditions that 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 where 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 file systems.

Since data is not affected by the settings in this file, you are encouraged to experiment and adjust the command settings to achieve the proper balance between archiving and staging requests, and the priorities of each preview request.

Following is a sample `preview.cmd` file that addresses the three conditions listed above:

```
# 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
```

Enforcing Stage Requests Prioritized over Archival Requests

The following example demonstrates one way to ensure that stage requests have priority over archival requests.

Make the following assumptions:

- Use the following `preview.cmd` file.
- Assume that several different requests are sitting in the queue for 100 seconds.
- Assume that 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.

Enforcing Archival Requests Prioritized When File system Over HWM

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 have at least one archive copy and are still disk-cache resident.)

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

```
hwm_priority = 500.0
```

Prioritizing Requests That Use a Specific Set of 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 as follows 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)
```

Complex Prioritization With Multiple File systems

Assuming there are two ASM file systems, the following `preview.cmd` file prioritizes requests where:

- 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 commands are:

```
lwm_priority =  -200.0

lhwm_priority =    0.0

hlwm_priority =    0.0
```

In this case, the other commands 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 (samfs2) from filling up. This might occur if 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 "`p`" flag is set.

```
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
```

Appendix A. Product Support & Enhancements

Introduction

This appendix describes StorageTek software support, how to report a problem to your StorageTek customer service engineer (CSE), and how StorageTek processes problems.

StorageTek Software Support

This part of the appendix relates to problems or questions regarding StorageTek software products, documentation, or support.

Customer Service Engineers

StorageTek uses customer service engineers (CSEs) to support its products. Your CSE is generally a specialist within your country or an industry market that provides support for StorageTek products. The CSEs provide help desk assistance and provide primary support on the product including taking the initial software support call and providing immediate problem and question resolution whenever possible.

StorageTek Support Center

When a CSE cannot provide a resolution, the problem is escalated by the CSE to the StorageTek support center, located at headquarters in Colorado. These secondary support calls require a higher level of expertise for problems that cannot be resolved by the Level-1 support representatives. If you have a CSE, the escalation of a problem to the StorageTek support center must be done through the CSE. Do not report a problem directly to StorageTek.

How to Report a Problem

This section describes the steps StorageTek suggests to report your problem to your CSE. If your CSE has a different procedure, please follow the CSE's procedure instead.

What to Do Before You Call

The first step to take when you encounter a problem or a technical question is to review the product documentation. It is possible that a solution or answer is provided in the documentation. If the documentation does not provides a solution, take the following steps before you call your support CSE.

Identify the Severity Level of the Problem

You must determine how critical the software problem is, based on the following descriptions of the severity levels.

The severity levels range from A to D, with severity level A being the most critical. The severity levels only apply to processing software problem incidents. All problems should be reported by E-mail if possible to avoid errors. We can help you in the most efficient way if you have E-mailed a description of the problem and `ftp`'ed the output of the `info.sh` script prior to telephoning. In the definitions that follow, "software problems" refers to both software and documentation problems.

The problem severity levels are:

- **Severity Level A -** The software product is non-operational, resulting in a critical system condition requiring immediate resolution. Support personnel may require continuous access to your resources until a workaround or resolution is provided. When reporting a Severity A incident, a telephone call after sending materials by E-mail is preferred. E-mail without a call is acceptable, but may delay a CSE or StorageTek response. StorageTek requires that any support calls at Severity Level A to StorageTek include availability of the customer system administrators to StorageTek.
- **Severity Level B -** The software product is operational, but with severely restricted functionality or system degradation. When reporting a Severity B incident, a telephone call after sending materials by E-mail is preferred, although E-mail alone is acceptable.
- **Severity Level C -** The software product is operational, with functional limitations or restrictions that are not critical to the overall system operations. When reporting a Severity C incident, E-mail is preferred.
- **Severity Level D -** Problems which have little, or no, impact on system operations. Severity D incidents should be reported only via E-mail.

Record Your Company Information

You will be asked to provide the following information:

- Your company name and customer number (for example, Company X, L0666).
- Your name and telephone number (also provide the name and number for an alternate contact, if possible). For Severity A or B, a pager number is also requested.
- Details of the software problem or technical question.
- Severity level of the software problem or a specific time and date by which you require a response to a technical question. See "Identify the Severity Level of the Problem" on page A-2. You will be given an incident number to track this support call.

Record Your Site and Configuration Information

You should have the following information accessible:

- The name of the software product, including the release level. For example, ASM 3.3.0-4.
- The names and releases of your system software, such as the operating systems or any other appropriate software. For example, Solaris 2.5.1.

Detail the Problem

Use the StorageTek Problem Identification Checklist on the following page to help you collect details about the software or documentation problem. You may make copies of the checklist. Collect this information for each problem or question.

Keep a record of your checklist responses in a convenient place for reference. Your CSE and the StorageTek software support center may request an answer to some or all of the questions.

Gather Diagnostic Outputs

During problem resolution, your CSE and the StorageTek support center may request that you provide specific diagnostic material. The output of the `info.sh` command provides us with a current snapshot of your system configuration as it relates to ASM. Since only limited log information is captured by this script, make sure you run `info.sh` *when the problem is occurring* or as soon a possible after it occurs. You should submit a separate copy of the `info.sh` output for each problem you report. The output should be FTP-ed to your CSE. There may be cases where StorageTek will direct you to specifically send information to the StorageTek site, but unless otherwise directed you should always work through your CSE. Please do not compress this ASCII file, unless you use the Solaris `compress`(1) utility. Your CSE wants to spend time resolving your problem, not spending time finding an obscure file decompression program.

Occasionally a problem is reported many hours or even days after the problem first occurred. Because ASM generates many log messages, a review of the log files during the problem period is important. The `info.sh` command only gathers the last 1000 log entries. If many log entries are generated after a problem occurs, a snapshot of the relevant time period may be needed for diagnosis.

Contact Support

When you have completed the steps described in “What To Do Before You Call” on page A-2, it is time to contact your CSE. Before contacting your CSE, make sure that you have collected all of the information on the Problem Identification Checklist.

StorageTek Problem Identification Checklist

For software problems, provide the following:

- Severity Level (see page A-2)
- Company name and customer number (see page A-2)
- Site and configuration information (see page A-3)
- What statement or command are you using?
- What are you expecting to happen, versus what is actually happening?
- What, if any, error messages are you receiving?
- When did you first notice the problem?
- Have you attempted this activity before? Was it successful?
- What has changed since the activity last operated correctly? For instance, was the software upgraded or have you changed the configuration?
- Is the problem reproducible? If so, under what conditions?
- Has the problem occurred before?
- If the problem does not occur consistently, describe the conditions under which the problem does and does not occur.
- What other information can you provide concerning this problem?

For documentation problems, provide the following:

- Severity Level (see page A-2)
- Company name and customer number (see page A-2)
- Site and configuration information (see page A-3)
- The document name, document number, and date of the publication.
- The number of the page that contains the problem.
- A description of the documentation problem.   Please be specific.
- Any other information that you can provide concerning this problem.

StorageTek Support Contacts

In situations in which your support contact is directly with StorageTek, then contact the StorageTek support center. See the table below.

StorageTek Support Center Hours of Operation

For standard 5 day, 8 hours per day support, the StorageTek support center hours of operations are Monday through Friday, 08:00 - 18:00 (Central Standard Time). This does not include StorageTek holidays.

Should you wish to contact the support center, use the following information:

Telephone (Domestic)	1 (800) 223-9834
Telephone (International)	Local StorageTek office
E-mail	salessupport@louisville.stortek.com

What StorageTek Does When Your CSE Reports a Problem

The StorageTek support center answers problems regarding our released software products. As problems are reported and analyzed, we work to clarify our documentation, fix our software, and implement design requests. The following process describes flow from reporting a problem to generating a fix.

Step 1: Log the Incident

When StorageTek receives a call or email, StorageTek support personnel will first assign an incident number, e.g., W711030018. The incident number is to be used for any communications regarding this specific problem. Multiple problems will usually generate multiple incident numbers. Be sure that, when you are responding to our StorageTek emails, you include the appropriate incident number in the E-mail subject line.

Step 2: Assign a Support Analyst to the Incident

The incident will be assigned to one of the support analysts in the StorageTek support center. This analyst takes the responsibility for working with your CSE to understand, analyze, and close out the problem. Not all incidents are necessarily answered immediately. If we cannot answer a problem right away, we will tell your CSE why and keep your CSE posted as we analyze the incident.

Step 3: Analyze the Problem Incident

Further analysis of the incident may be required. If so, the support analyst will work with the proper StorageTek personnel to try and resolve the problem.

Step 4: Request Additional Information

During problem resolution, the support analyst may request additional information from you through your CSE. The analyst will refer to the problem using the assigned incident number. You should remember to use this number upon response as well.

Step 5: Close Out the Incident Report

When the problem is resolved, StorageTek will close the incident with a response to you through your CSE. An incident is closed when a) A question is answered; b) A problem that you encountered is fixed in a release that may require you to upgrade your software; c) StorageTek has analyzed the problem and determined that there is a problem in our software that needs to be fixed.

In this last case, we will close the incident but will open a software problem report as described in the next step.

Step 6: StorageTek Opens a Problem Report

If an incident is deemed to be an error in our software, the StorageTek support analyst will open an internal software problem. Problem reports are tracked in an StorageTek internal database and assigned to software developers for further analysis and to be fixed.

Step 7: Take Corrective Action

The software developer will work to fix the problem. Although we do work to close all of our software problem reports in a timely manner, no guarantees are made as to when these problems might be resolved.

Step 8: Integrate and Test

The software fix is integrated into an internal software release and tested before it is released. Sometimes we may request a customer to help in the testing process by running a fix at their site. This is done for situations where a unique set of circumstances exist for the problem only at the customer site.

Step 9: Identify a Release in Which to Package the Software Fix

Finally, the software fix is integrated into one or more releases of the software product. StorageTek's software fixes are packaged in releases indicated by the last digit in the release number. For example, ASM 3.2.0-9 is the ninth bugfix edition of 3.2.0.

Appendix B. Disaster Recovery

Introduction

This appendix shows various methods of reading data from archive media (both MO and tape) both with and without using the ASM software. Such information is valuable when creating or executing a disaster recovery plan.

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.

ASM Capabilities

A typical `archiver(1M)` log file entry consists of a single line with fields delimited by spaces. An example entry and a description of the fields is shown below:

```
A 96/05/01 15:41:31 mo v1 set_1.1 d2e.b samfs1 176942 161.5
dir1/dir3/file0
```

Field	Description	Example

1	Indicates activity	`A = archive`
		`U = unarchive`
		`1-16`

2	Date (yy/mm/dd)	`99/01/01`
3	Time (hh:mm:ss)	`15:41:31`
4	Media type	`mo`
5	VSN	`v1`
6	Archive set name.copy	`set_1.1`
7	Tarfile position on medium (hexadecimal)	`d2e`

8	File offset in tarfile (in units of 512 bytes)	`b`
9	Filesystem name	`samfs1`
10	Inode number	`161`
11	Inode generation	`5`
12	File length in bytes. If volume overflow is used, this indicates section length in bytes.	`176942`
13	Pathname	`dir1/dir3/file0`

The "position on medium" can be used with the `request`(1) command with the `-p` option to position at the beginning of the tar header for the file. Use hexadecimal notation starting with 0x. For example, to position to the above 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 offsets us to the beginning of the tar header for the exact file; the log file simply shows the beginning of the tar file as a whole.

"**.b**" is the offset * 512 for the logical offset to the beginning of the tar header for the exact file. For this case the file is 5632 bytes from the beginning of the tar file.

```
# 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 next two sections show how to recover this file's data.

Note: The invocations of `gnutar` in this appendix use the "t" option for demonstrative purposes. If you actually wish to extract the data from the tar files, replace the "t" option with "x". For more information on `guntar`, run the following:

```
# /opt/LSCsamfs/sbin/gnutar --help
```

Here's an example of using the log file information, the `request`(1) command and `gnutar`(1) to reference a file on optics. We get the VSN and position from the log file, use them to construct a `request`(1) command which creates a temporary file **`/sam/xxx`**, which we then pass to `gnutar`(1).

```
#  request -b -p 0xd2e mo v1 /sam/xxx
#  gnutar 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
```

Here's an example of using the log file, `dd` and `gnutar` to reference the file:

```
#  dd if=/dev/samst/c0t1u0 bs=1k iseek=3374 of=/tmp/junk
count=10
dd: read error: I/O error    <---- This is OK!
8+0 records in
8+0 records out
#  gnutar 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
```

Now let's consider the case where data is resident on tape. Here's 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
```

(for DLT).

```
# 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 kbytes) you must use the block size in bytes divided by 512 instead of "32" for the `gnutar` "b" parameter. You can see the tape block size by mounting the tape and observing the `samu(1M)` "t" display.

Here's how the log file information, the `request`(1) command and `gnutar`(1) can be used to reference a file on tape:

```
# request -p 0x286 lt YYY /sam1/xxx
# gnutar 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
#
```

Again, note that we've ignored the "**.1324f**" part of the position information which was shown in the `sls -D` output, but not in the log file. Had we used this offset into the tar file, we could have positioned directly to the data. As it was, we saw some extra files before the one we were interested in.

You can also recover data from tape without using the `request`(1) command. If you are currently running ASM, shut it down by unmounting all the ASM file systems, and typing **`kill -INT`** *`pid`*, where *pid* is the process-id of the `sam-init` program. This will keep ASM from attempting to use the tape from which we are recovering data.

Make sure ASM doesn't 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.

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

The first "file" is the ANSI label. Here we display it, but you can also skip over it if you wish:

```
# 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's the first `tar` file:

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

Here's the next `tar` file. Note that I've archived all files twice, so this is the second copy:

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

Next, we see two copies of another file:

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

Then we hit the end of the tape:

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

Appendix C. Manual Pages

Introduction

This appendix contains a printed version of the online manual pages provided with the ASM 3.3 release.

Glossary

addressable storage	The storage space encompassing on-line, near-line, and off-line storage that is user-referenced through the ASM file system.
archiver	The archive program that automatically controls the archiving of files to removable media.
archive storage	Copies of file data that have been created on removable media for long-term off-line storage.
ASM	The StorageTek Storage and Archive Manager File System. ASM controls the access to all files stored and all devices configured in the Master Configuration File.
ASM-QFS	ASM-QFS is an ASM-compatible file system that offers a high speed, standard Unix file system interface to users and administrators. It can be used as a stand-alone file system or with ASM. It uses many of the commands available in the ASM command set as well as standard UNIX file system commands.
audit (full)	The process of reading the USNs from each piece of media in a robot. For non-tape media, the capacity and space information is determined and entered into the robot's catalog.
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.
data space	The portion of a collection of files that is the actual data information.
DAU	Disk Allocation Unit. A basic unit of on-line storage. ASM uses several sizes. The small DAU is 4 kilobytes and the large DAU is 16, 32, or 64 kilobytes. The available DAU size pairs are 4/16, 4/32, and 4/64.

device logging	An ASM feature that provides device-specific error information used to analyze device problems.
device scanner	A function of ASM that periodically monitors the presence of all ASM manually mounted removable devices and detects the presence of mounted media that may be requested by a user.
devicetool	ASM administrative tool with 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 direct I/O option sets the direct I/O attribute for a file or directory. If applied to a directory, the direct I/O attribute is inherited.
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 near-line file can be accessed directly from the archive media and need not be staged on-line for access.
directory	A file data structure pointing 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.
family device set	A storage device that is represented by a group of independent devices, such as a collection of disks or the drives mounted within a library device.
FDDI	Fiber Distributed Data Interface. FDDI is a 100MB per second fiber optic LAN.
file-specific commands	Commands that follow global commands and begin with "`fs =` ". File-specific commands apply until the next "`fs =` " line or the end of file is encountered. If multiple commands impact a file system, the file system-specific commands override the global commands.

file system A hierarchical collection of files and directories. The file system is a common contact point for the user and influences the user's view of the operating system.

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 block allocated to a file that allows the size of the file to grow exponentially with each level of indirectness. Each indirect block either points to a series of data blocks for the file or a series of indirect blocks allocated to the next level. ASM has up to three levels of 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, such as ownership and where the file is allocated on the disk system.

inode file A special file (`.inodes`) on ASM that contains the inode structures for all files resident in the file system. ASM inodes are 512 bytes.

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 A robotically-controlled device designed to automatically load and unload removable media without operator intervention. A library consists of three parts: a robot that moves media to and from storage slots and drives.

mcf Master Configuration File. A file that is read at initialization time and defines the device topology for an ASM server.

media Tapes or disks.

media recycling The process of recycling or reusing archive media with low use (i.e., archive media with few active files).

medium A tape or disk.

mirror writing The process of maintaining two copies of a file on distinct 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.
near-line storage	Removable storage that requires robotic mounting before it can be accessed. Near-line storage is usually less expensive than on-line storage, but requires a somewhat longer access time.
Network-controlled library	A network-controlled library such as the StorageTek (STK), Grau, or IBM, is controlled using a software package supplied by the vendor. ASM interfaces with the vendor software using a StorageTek media changer daemon specifically designed for the library.
NFS	Network File System. A standard protocol that allows a UNIX file system to be remotely mounted via a network.
off-line storage	Storage that requires operator intervention for loading.
off-site storage	Storage that is remote from the ASM server and used for disaster recovery.
on-line storage	Storage that is immediately available (e.g., disk storage).
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 protects against data loss from a single disk failure and provides a fault-tolerant disk environment.
recycler	The ASM program that reclaims the space 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 files properties and then summing the results.
releaser	The ASM program that makes disk cache space available by identifying archived files and releasing their disk cache copies.
removable media file	A special type of user file used to access removable media, such as magnetic tape or optical disk.
robot	The portion of a library that moves media between storage slots and drives.

robottool
: ASM administrative tool with graphical user interface for viewing and managing libraries.

RPC
: Remote Procedure Calls. The underlying data exchange mechanism used by NFS to implement custom network data servers.

samfsdump
: 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` utility, but it does not copy data.

samfsrestore
: Restores a control structure dump.

samtool
: ASM administrative tool with graphical user interface for starting `robottool`, `devicetool`, and `previewtool`.

SCSI
: Small Computer System Interface. An electrical communication specification commonly used for peripheral devices.

SCSI-attached Library
: A library connected directly to a server using SCSI. These libraries are controlled directly by ASM using the SCSI standard for media changers.

staging
: The process of copying a near- or off-line file from its archive storage back to on-line storage.

storage family set
: A set of disks that are collectively represented by a single disk family device.

storage slots
: Locations inside a library where volumes are stored when not being used in a drive. The contents of the storage slots are kept in the library's catalog.

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 file/data recording format used by ASM 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).

volume
: A piece of media.

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 media.

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

Symbols

A

B

C

D

E

F

G

H

I

L

M

O

P

R

S

T

U

V